A peer-reviewed version of this preprint was published in PeerJ on 18 November 2019.

View the peer-reviewed version (peerj.com/articles/cs-237), which is the preferred citable publication unless you specifically need to cite this preprint.

Nardone D, Ciaramella A, Staiano A. 2019. A Sparse-Modeling Based Approach for Class Specific Feature Selection. PeerJ Computer Science 5:e237 https://doi.org/10.7717/peerj-cs.237

A Sparse-Modeling based approach for Class-Specific feature selection

Davide Nardone [Corresp., 1], **Angelo Ciaramella** [1], **Antonino Staiano** [1]

[1] Dipartimento di Scienze e Tecnologie, Universita` degli Studi di Napoli "Parthenope", Naples, Italy

Corresponding Author: Davide Nardone
Email address: davide.nardone@live.it

In this work, we propose a novel Feature Selection framework, called Sparse-Modeling Based Approach for Class Specific Feature Selection (SMBA-CSFS), that simultaneously exploits the idea of Sparse Modeling and Class-Specific Feature Selection. Feature selection plays a key role in several fields (e.g., computational biology), making it possible to treat models with fewer variables which, in turn, are easier to explain, by providing valuable insights on the importance of their role, and might speed the experimental validation up. Unfortunately, also corroborated by the no free lunch theorems, none of the approaches in literature is the most apt to detect the optimal feature subset for building a final model, thus it still represents a challenge. The proposed feature selection procedure conceives a two steps approach: (a) a sparse modeling-based learning technique is first used to find the best subset of features, for each class of a training set; (b) the discovered feature subsets are then fed to a class-specific feature selection scheme, in order to assess the effectiveness of the selected features in classification tasks. To this end, an ensemble of classifiers is built, where each classifier is trained on its own feature subset discovered in the previous phase, and a proper decision rule is adopted to compute the ensemble responses. In order to evaluate the performance of the proposed method, extensive experiments have been performed on publicly available datasets, in particular belonging to the computational biology field where feature selection is indispensable: the acute lymphoblastic leukemia and acute myeloid leukemia, the human carcinomas, the human lung carcinomas, the diffuse large B-cell lymphoma, and the malignant glioma. SMBA-CSFS is able to identify/retrieve the most representative features that maximize the classification accuracy. With top 20 and 80 features, SMBA-CSFS exhibits a promising performance when compared to its competitors from literature, on all considered datasets, especially those with a higher number of features. Experiments show that the proposed approach might outperform the state-of-the-art methods when the number of features is high. For this reason, the introduced approach proposes itself for selection and classification of data with a large number of features and classes.

A Sparse-Modeling Based Approach for Class-Specific Feature Selection

Davide Nardone[1], **Angelo Ciaramella**[1], **and Antonino Staiano**[1]

[1]**Dipartimento di Scienze e Tecnologie, Università degli Studi di Napoli "Parthenope", Centro Direzionale, Isola C4, 80143, Naples, Italy**

Corresponding author:
Davide Nardone[1]

Email address: davide.nardone@studenti.uniparthenope.it

ABSTRACT

In this work, we propose a novel Feature Selection framework, called Sparse-Modeling Based Approach for Class Specific Feature Selection (SMBA-CSFS), that simultaneously exploits the idea of *Sparse Modeling* and *Class-Specific Feature Selection*. Feature selection plays a key role in several fields (e.g., computational biology), making it possible to treat models with fewer variables which, in turn, are easier to explain, by providing valuable insights on the importance of their role, and might speed the experimental validation up. Unfortunately, also corroborated by the no free lunch theorems, none of the approaches in literature is the most apt to detect the optimal feature subset for building a final model, thus it still represents a challenge. The proposed feature selection procedure conceives a two steps approach: (a) a sparse modeling-based learning technique is first used to find the best subset of features, for each class of a training set; (b) the discovered feature subsets are then fed to a class-specific feature selection scheme, in order to assess the effectiveness of the selected features in classification tasks. To this end, an ensemble of classifiers is built, where each classifier is trained on its own feature subset discovered in the previous phase, and a proper decision rule is adopted to compute the ensemble responses. In order to evaluate the performance of the proposed method, extensive experiments have been performed on publicly available datasets, in particular belonging to the computational biology field where feature selection is indispensable: the *acute lymphoblastic leukemia* and *acute myeloid leukemia*, *the human carcinomas*, the *human lung carcinomas*, the *diffuse large B-cell lymphoma*, and the *malignant glioma*. SMBA-CSFS is able to identify/retrieve the most representative features that maximize the classification accuracy. With top 20 and 80 features, SMBA-CSFS exhibits a promising performance when compared to its competitors from literature, on all considered datasets, especially those with a higher number of features. Experiments show that the proposed approach might outperform the state-of-the-art methods when the number of features is high. For this reason, the introduced approach proposes itself for selection and classification of data with a large number of features and classes.

INTRODUCTION

Feature Selection (FS) is the process of selecting a subset of relevant features for use in model construction. FS plays a key role in computational biology, for instance, microarray data analysis involves a huge number of genes w.r.t. a small number of samples, and effectively identifying the most significant differentially expressed genes under different conditions is prominent (Xiong et al., 2001). The selected genes are very useful in clinical applications such as recognizing diseased profiles (Calcagno et al., 2010; Staiano et al., 2013; Di Taranto et al., 2015; Camastra et al., 2015), nonetheless, because of its high costs, the number of experiments that can be used for classification purposes is usually limited so that the small number of samples, compared to the large number of genes in an experiment, gives rise to the *Curse of Dimensionality* problem (Friedman et al., 2001), which challenges the classification as well as other data analysis tasks (Staiano et al., 2004; Ciaramella et al., 2008). Furthermore, microarray data are usually not immune from several issues, such as sensitivity, accuracy, specificity, reproducibility of results, and noisy data (Draghici et al., 2006). For these reasons, it is unsuitable using microarray data as they are, but, after several corrections, select the relevant genes by FS approaches and, for instance, validate the results using Real-Time PCR (Xiong et al., 2001).

Taking a look at the literature, by *googling* the keyword "*feature selection*", one gets lost in an ocean of techniques (the reader might refer to classical reviews in (Saeys et al., 2007) and (Guyon and Elisseeff, 2003) on the topic), often designed to tackle a specific data set. The reasons for the abundance of techniques are in the heterogeneity of the available scientific data sets and also by the limitations dictated by *no free lunch theorems* (Wolpert and Macready, 1997), determining the existence of no general-purpose technique which well suites to a plethora of different kind of data. A typical taxonomy organizes FS techniques (Jović et al., 2015) in three main categories, namely *filter*, *wrapper* and *embedded* methods, whose belonging algorithms select a single feature subset from a complete list of features. Another perspective instead, divides FS techniques in two classes, namely, Traditional Feature Selection (TFS) for all classes (that includes filter, wrapper and embedded methods mentioned so far), and Class-Specific Feature Selection (CSFS) (Fu and Wang, 2002). Usually, a TFS algorithm selects one subset of features for all classes although it might be not the best one for some class, thus leading to undesirable results. Differently, a CSFS policy permits to select a distinct subset for each class, and it can use any traditional *feature selector*, for choosing, given the set of classes of a classification problem, one distinct grouping of features for every class. Depending on the type of the feature selector, the overall process may slightly change. Nevertheless, it is worth pointing out that a CSFS scheme heavily depends on the use of a specific classifier, while its use should be independent of both the classifier of the classification step and the feature selector strategy. To this end, a General Framework CSFS has been proposed in (Pineda-Bautista et al., 2011) which allows using any traditional feature selector as well as any classifier, consisting of four stages (the reader may refer to Methods section later on).

In this paper, on the basis of the general framework for CSFS, we propose a novel strategy to FS, namely a Sparse-Modeling based approach for Class-Specific Feature Selection, consisting of a two-steps procedure. Firstly, a sparse modeling based learning technique is used to find the best subset of features for each class of the training set. In so doing, it is assumed that a class is represented by using a subset of features, called *representatives*, such that each sample in a specific class, can be described as a linear combination of them. Secondly, the discovered feature subsets are fed to a class-specific feature selection scheme in order to assess the effectiveness of the selected features in classification task. To this end an ensemble of classifiers is built by training a given classifier, one for each class, on its own feature subset, i.e., the one discovered in the previous step, and a proper decision rule is adopted to compute the ensemble responses. In this way, the dilemma of choosing specific TFS strategy and classifiers in the CSFS framework is effectively mitigated.

METHODS

The sparse-modeling based approach for class-specific feature selection, is based on the concepts of sparse modeling and class-specific feature selection that need to be properly introduced.

Sparse Modeling fundamentals

An active developing field of statistical learning is around the notion of sparsity (Tibshirani, 1994; Ciaramella and Giunta, 2016). A Sparse Model (SM) is a model that can be much easier to estimate and

interpret than a dense model. The sparsity assumption allows extracting meaningful features from large datasets. Aim of the first phase of the proposed approach is to use a sparse modeling for finding data representatives without data transformation and to be performed directly in the data space. In other words, we wish to find a ranking of the most representatives features that best reconstruct the data collection. Most approaches are based on a l_1-norm regularization (e.g, LASSO (Tibshirani, 1994), Sparse Dictionary Learning (Elhamifar et al., 2012)). Formally, given a set of features in $\mathbb{R}^m$ arranged as columns of a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, the task is to find representative features given a fixed feature space belonging to a collections of data points (see (Mairal et al., 2008; Aharon et al., 2006; Engan et al., 1999; Jolliffe, 1986; Ramirez et al., 2010)). That task can conveniently be described in the *Dictionary Learning* (DL) framework, where the aim is to simultaneously learn a compact dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ and coefficients $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_n] \in \mathbb{R}^{k \times n}$, with $k \ll n$, that can well represent collections of data points (Ciaramella et al., 2016). The best representation of the data is obtained by minimizing the following objective function

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|_2^2 = \|\mathbf{X} - \mathbf{D}\mathbf{C}\|_F^2 \tag{1}$$

w.r.t. the dictionary $\mathbf{D}$ and the coefficient matrix $\mathbf{C}$, subject to appropriate constraints.

However, the dictionary learned atoms almost never correspond to the original feature space (Aharon et al., 2006; Ramirez et al., 2010; Mairal et al., 2009). In order to find a subset of features that best represent the entire feature space, the optimization problem in 1 is reformulated forcing the dictionary $\mathbf{D}$ to be the data matrix $\mathbf{X}$ (Elhamifar et al., 2012):

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{X}\mathbf{c}_i\|_2^2 = \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2, \tag{2}$$

where F is the Frobenius norm. Equation 2 is minimized w.r.t the coefficient matrix $\mathbf{C} \triangleq [\mathbf{c}_1, \ldots, \mathbf{c}_n] \in \mathbb{R}^{n \times n}$, subject to additional constraints. In other words, the *reconstruction error* of each feature component is minimized by linearly combining all components of the feature space. To choose $k \ll n$ representatives involved in the linear reconstruction of the each component in (2), the following constraint is added to the model

$$\|\mathbf{C}\|_{0,q} \leq k, \tag{3}$$

where the mixed ℓ_0/ℓ_q norm is defined as $\|\mathbf{C}\|_{0,q} \triangleq \sum_{i=1}^{N} I(\left\|\mathbf{c}^i\right\|_q > 0)$, $\mathbf{c}^i$ denotes the i-th row of $\mathbf{C}$, and $I(\cdot)$ denotes the indicator function. In a nutshell, $\|\mathbf{C}\|_{0,q}$ counts the number of nonzero rows of $\mathbf{C}$. The indices of the nonzero rows of $\mathbf{C}$ correspond to the indices of the columns of $\mathbf{X}$ which are chosen as the representative features. Since the aim is to select $k \ll n$ representatives features that can reconstruct each feature of the $\mathbf{X}$ matrix up to a fixed error, the optimization problem to solve is

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 \\ \text{subject to} \quad & \|\mathbf{C}\|_{0,q} \leq k, \mathbf{1}^T\mathbf{C} = \mathbf{1}^T \end{aligned} \tag{4}$$

where $\mathbf{1}^T\mathbf{C} = \mathbf{1}^T$ is the affine constraint for selecting representatives that are invariant w.r.t. a global translation of the data (as requested by dimensionality reduction methods). This is an NP-hard problem as it implies a combinatorial calculation over every subset of the k columns of $\mathbf{X}$. Therefore, relaxing ℓ_0 to ℓ_1 norm, the problem becomes

$$\begin{aligned} \underset{\mathbf{C}}{\text{minimize}} \quad & \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 \\ \text{subject to} \quad & \|\mathbf{C}\|_{1,q} \leq \tau, \mathbf{1}^T\mathbf{C} = \mathbf{1}^T \end{aligned} \tag{5}$$

where $\|\mathbf{C}\|_{1,q} \triangleq \sum_{i=1}^{N} \left\|\mathbf{c}^i\right\|_q$ is the sum of the ℓ_q norms of the rows of $\mathbf{C}$ and $\tau > 0$ is an appropriate chosen parameter. The solution of the optimization (5) not only provides the representative features as the nonzero rows of the $\mathbf{C}$, but also provides information about the ranking of the selected features. More

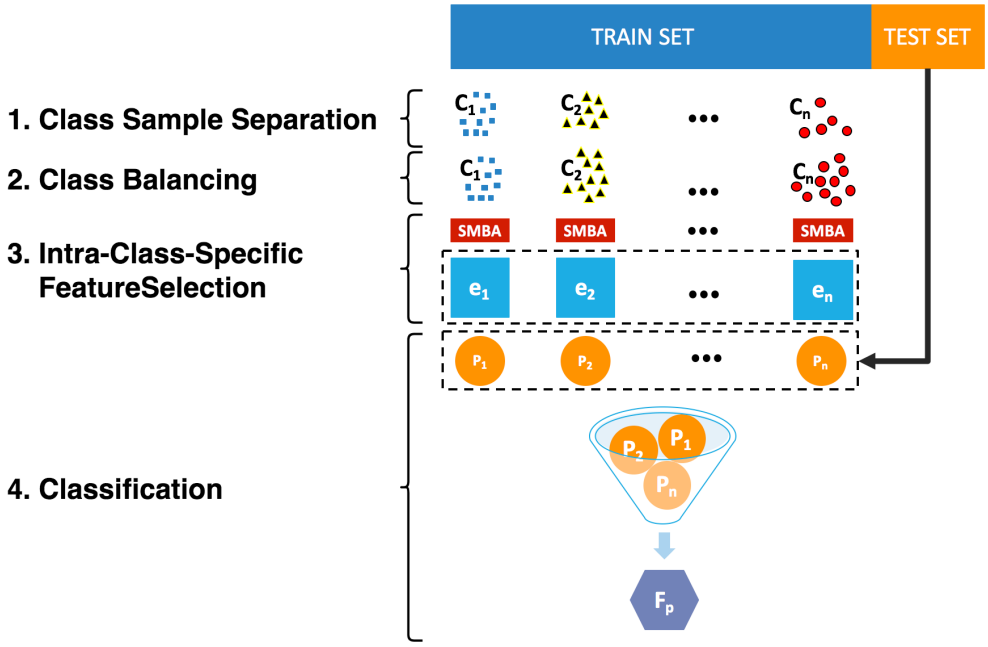

Figure 1. A Sparse-Modeling Based approach for Class-Specific Feature Selection.

precisely, a representative that has higher ranking takes part in the reconstruction process more than the others, hence, its corresponding row in the optimal coefficient matrix $\mathbf{C}$ has many nonzero elements with large values. Conversely, a representative with lower ranking takes part in the reconstruction process less than the others, hence, its corresponding row in $\mathbf{C}$ has a few nonzero elements with smaller values. Thus, the k representative features $x_{i1},\ldots,x_{ik}$ are ranked as $i_1 \geq i_2 \geq \cdots \geq i_k$, whenever for the corresponding rows of $\mathbf{C}$ one gets

$$\left\|\mathbf{c}^{i_1}\right\|_q \geq \left\|\mathbf{c}^{i_2}\right\|_q \cdots \geq \left\|\mathbf{c}^{i_k}\right\|_q, \tag{6}$$

From a practical point of view, the optimization problem (5) can be expressed by using the Lagrange multipliers

$$\underset{\mathbf{C}}{\text{minimize}} \quad \frac{1}{2}\left\|\mathbf{X}-\mathbf{X}\mathbf{C}\right\|_F^2 + \lambda\left\|\mathbf{C}\right\|_{1,q} \quad \text{subject to} \quad \mathbf{1}^T\mathbf{C} = \mathbf{1}^T. \tag{7}$$

In practice, the algorithm is implemented using an Alternating Direction Method of Multipliers (ADMM) optimization framework (Boyd et al., 2011). In particular, the features of a given dataset are obtained considering representatives of small pairwise coherence features as in a sparse dictionary learning method. It is worth observing the resemblance with the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1994). LASSO consists of an approach to regression analysis that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretation ability of the statistical model it produces. Recall that the objective of LASSO, in its basic form, is to solve

$$\begin{aligned} &\underset{\beta}{\text{minimize}} && \frac{1}{N}\left\|y-\mathbf{X}\beta\right\|_2^2 \\ &\text{subject to} && \left\|\beta\right\|_1 \leq t, \end{aligned} \tag{8}$$

where $y = [y_1,\ldots,y_N]$ is the N-dimensional vector of outcomes, $\mathbf{X}$ the covariate matrix, t is a free parameter that determines the amount of regularization and β is the sparse vector to estimate.

From Equation 8, one can observe that a sparse matrix can be estimated as in equation 7 by considering $\mathbf{X}$ itself as outcome and adding the affine constraint. In the following, the LASSO will be used for classification tasks, adopting a sigmoid function, as it will be described in the experimental setup.

A Sparse-Modeling Based Approach for Class-Specific Feature Selection

A General Framework for Class-Specific Feature Selection (GF-CSFS) is described in (Pineda-Bautista et al., 2011). The proposed Sparse-Modeling Based Approach for Class-Specific Feature Selection (SMBA-CSFS) tries to best represent each class-sample set of an input dataset by only using few representatives features. More specifically, the method is made up of the following steps:

1. **Class-sample separation:** Unlike the GF-CSFS, SMBA-CSFS does not employ the *Class binarization* stage to transform a c-class problem into c binary problems, instead it just uses a simple *Class-sample separation*. It simply consists of differentiating the samples among all the classes of the training set for a given dataset into several disjoint sets/configurations of samples, one for each class (See Fig. 1).

2. **Class balancing:** Once the class sample set of the training set has been split apart, it may be possible that each class-subset results unbalanced. Therefore, the SMOTE (Chawla et al., 2002) re-sampling method is applied to balance each class-subset.

3. **Intra-Class-Specific feature selection:** The *sparse-modeling based approach* is used for retrieving, minimizing equation 7, the most representative features for each class-sample set of the training set that best represent/reconstruct the whole class of objects. In doing so, the approach takes advantage of the intra-class properties for selecting the best feature subset (describing each class) which is used to improve the classification accuracy against TFS and GF-CSFS.

4. **Classification:** Since the training set gets split into different class-sample subsets, we embraced the idea of using a wise-ensemble procedure for training a classification model for discriminating new incoming instances. As in (Pineda-Bautista et al., 2011), given a class c_i, a classifier e_i is trained on the original dataset only using the selected features for c_i, for $i = 1,\ldots,c$. Overall, a classifier ensemble $E = \{e_1,\ldots,e_c\}$ is constructed. In order to classify a new instance O through the ensemble, the natural dimension of O needs to be lowered to the dimension d_i of the classifier $e_i, i = 1\ldots,c$. This way, for determining to which class O belongs to, an *ad-hoc majority rule* is used:

   (a) If a classifier outputs the same class for which the features, used for e_i training, were selected, i.e., the e_i output is c_i, then O belongs to c_i. In case of a tie, i.e., when several classifiers respond c_i, a *majority vote* is needed among all classifiers to determine the class of O. If still a tie occurs, O will belong to the class that received more votes among the tied classes.

   (b) If no classifier outputs the class whose selected features are used for e_i training, O belongs to the class winning the majority voting. If there is a tie, then O will belong to the class that received more votes among the tied classes.

EXPERIMENTAL RESULTS

In the experiments, the SMBA-CSFS performance have been assessed on eight publicly available microarray datasets. The classifier used to determine the goodness of the selected feature subsets are a Support Vector Machine (SVM) with a linear kernel and parameter $C = 1$, a Naive Bayes, a K-Nearest Neighbors (KNN) using $k = 5$, and a Decision Tree.

Datasets Description

In order to validate the introduced approach, a number of datasets exemplifying the typical data processing in the biological field are used in the experiments. In the following, a brief description of all datasets employed in the experiments.

1. The **ALLAML** dataset (Golub et al., 1999) contains in total 72 samples in 2 classes, ALL and AML, which have 47 and 25 samples, respectively. Every sample contains $7,129$ gene expression values.

2. The **LEUKEMIA** dataset (Golub et al., 1999) contains in total 72 samples in 2 classes: acute lymphoblastic and acute myeloid. From $7,129$ genes, the baseline genes were cut off before further analysis. The number of genes that are used in the multiclass classification task is $7,070$.

3. The **CLL_SUB_111** dataset (Haslinger et al., 2004) has gene expressions from high density oligonucleotide arrays containing genetically and clinically distinct subgroups of B-cell chronic lymphocytic leukemia (B-CLL). The dataset consists of $11,340$ attributes, 111 instances and 3 classes.

4. The **GLIOMA** dataset (Nutt et al., 2003) contains in total 50 samples in 4 classes: cancer glioblastomas, non-cancer glioblastomas, cancer oligodendrogliomas and non-cancer oligodendrogliomas, which have 14, 14, 7, 15 samples, respectively. Each sample has $12,625$ genes. After a preprocessing, the dataset has been shrunk to 50 samples and $4,433$ genes.

5. The **LUNG** dataset (Bhattacharjee et al., 2001) contains in total 203 samples in 5 classes, adenocarcinomas, squamous cell lung carcinomas, pulmonary carcinoids, small-cell lung carcinomas and normal lung, with $139, 21, 20, 6, 17$ samples, respectively. The genes with standard deviations smaller than 50 expression units were removed (the interested reader may refer to (Bhattacharjee et al., 2001) for details) getting a dataset with 203 samples and $3,312$ genes.

6. The **LUNG_DISCRETE** dataset (Peng et al., 2005) contains 73 samples in 7 classes where, each sample consists of 325 gene expressions. The cardinalities of each sample in the LUNG_DISCRETE dataset are $6, 5, 5, 16, 7, 13, 21$, respectively.

7. The **DLBCL** dataset (Alizadeh et al., 2000) is a modified version of the original DLBCL dataset. It consists of 96 samples in 9 classes, where each sample is defined by the expression of $4,026$*genes*. The cardinalities of each sample in the DLBCL dataset are $46, 10, 9, 11, 6, 6, 4, 2, 2$, respectively.

8. The **CARCINOM** dataset (Su et al., 2001) contains 174 samples in 11 classes, prostate, bladder/ureter, breast, colorectal, gastroesophagus, kidney, liver, ovary, pancreas, lung adenocarcinomas and lung squamous cell carcinoma, with $26, 8, 26, 23, 12, 11, 7, 27, 6, 14, 14$ samples, respectively. After a preprocessing as described in (Yang et al., 2006), the dataset has been shrunk to 174 samples and $9,182$ genes.

All datasets have been originally downloaded from the following source, migrated at later time at the following data repository (Nardone et al., 2019a). All the information about the datasets are summarized in Table 1.

Experiment Setup

To validate the effectiveness of the SMBA-CSFS model, it has been compared against several TFS and the GF-CSFS proposed in (Pineda-Bautista et al., 2011). SMBA-CSFS is firstly compared against TFS methods and, since the framework in (Pineda-Bautista et al., 2011) can use any TFS method as base for doing CSFS, some experiments using both filter and wrapper methods (injection process) were made. In addition, the accuracy results were also compared against those obtained on the basis of all the features (BSL). The following TFS methods have been chosen for comparing purposes:

- **LASSO** (Tibshirani, 1994): It involves penalizing the absolute size of the regression coefficients and is usually used for creating parsimonious models in presence of a *large* number of features. The model implemented is a modified version of classical LASSO, adapted for classification purposes. In particular, in Equation 8, the product $\mathbf{X}\boldsymbol{\beta}$ is transformed by a sigmoid function in order to address the classification problem.

- **EN** (Zou and Hastie, 2005): Elastic Net is a hybrid of ridge regression and LASSO regularization. Like lasso, Elastic Net can generate reduced models by generating zero-valued coefficients. Experimental studies have suggested that the Elastic Net technique can outperform LASSO on data with highly correlated features. As for LASSO, a modified version adapted for classification purposes has been implemented.

- **RFS** (Nie et al., 2010): Robust Feature Selection method is a sparse based-learning approach for feature selection which emphasizes the joint $\ell_{2,1}$ norm minimization on both loss and regularization function.

- **ls-$\ell_{2,1}$** (Tang et al., 2014): ls-$\ell_{2,1}$ is a supervised sparse feature selection method. It exploits the $\ell_{2,1}$-norm regularized regression model for joint feature selection, from multiple tasks where the *classification objective function* is a quadratic loss.
- **ll-$\ell_{2,1}$** (Tang et al., 2014): ll-$\ell_{2,1}$ is a supervised sparse feature selection method which uses the same concept of ls-$\ell_{2,1}$ but instead uses a *logistic loss* as *classification objective function*.
- **Fisher** (Gu et al., 2012): Fisher is one of the most widely used supervised filter feature selection methods. It selects each feature as the ratio of inter-class separation and intraclass variance, where features are evaluated independently and, the final feature selection occurs by aggregating the m top ranked ones.
- **Relief-F** (Kira and Rendell, 1992; Kononenko, 1994): Relief-F is an iterative, randomized and supervised filter approach that estimates the quality of the features according to how well their values differentiate data samples that are near to each other; it does not discriminate among redundant features and performance decreases with few data.
- **mRmR** (Peng et al., 2005): Minimum-Redundancy-Maximum-Relevance is a mutual information filter based algorithm which selects features according to the maximal statistical dependency criterion.
- **MI** (Kraskov et al., 2004; Ross, 2014): Mutual Information is a non-negative value, which measures the dependency between the variables. Features are selected in a univariate way. The function relies on nonparametric methods based on entropy estimation from k-nearest neighbors distances.
- **SMBA**: Sparse-Modeling Based Approach is nothing else that our SMBA-CSFS model but that only take into account the SDL strategy for selecting a subset of features considering all the classes in the feature selection process.

We pre-processed all the datasets by using the *Z-score* (Kreyszig, 2010) normalization. To fairly compare the considered supervised feature selection methods, we have firstly tuned the parameters for all methods by using a "grid-search" strategy (Tang et al., 2014) and finally, for evaluating the performance of all the methods, it has been considered a number of features ranging from 1 to 300, performing a 5-fold CV to report the average results along with the standard deviations (STD).

The evaluation metric used for assessing the classification performance among all the methods is the *accuracy* (ACC). It's defined as follows:

$$ACC(y,\hat{y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} 1(\hat{y}_i = y_i) \tag{9}$$

where y_i and $\hat{y}_i$ are, respectively, the ground truth and the predicted label of the i-th samples and, $n_{samples}$ is the number of samples of the testing set. Obviously, a larger *ACC* indicates a better performance.

DISCUSSION

The experiments have been performed on a workstation with a dual Intel(R) Xeon(R) 2.40GHz and 64GB RAM. The developed code is available at (Nardone et al., 2019b).

For all comparisons, we computed the average ACC along with its STD accuracy using the top 20 and top 80 features. In case of a tie among methods, we have considered the best achieved accuracy with a fewer number of features.

For the sake of readability, all the results presented here account only for the SVM classifier, since the performance prove that the proposed approach is a little sensitive to the choice of a specific classifier (indeed, the performance of each classifier are rather comparable). Nevertheless, the interested reader may refer to the supplementary material for details on additional results concerning all the used classifiers. The experimental results on 5-CV for the SVM classifier are summarized in the Tables 2-5. Figures 2 and 3 show the classification accuracies of all the ten feature selection methods on the eight considered data sets.

We compared the performance of our method against TFS methods (see Tables 2-3) and GF-CSFS framework (see Tables 4-5). SMBA-CSFS is able to better discriminate among the classes of the LUNG_C,

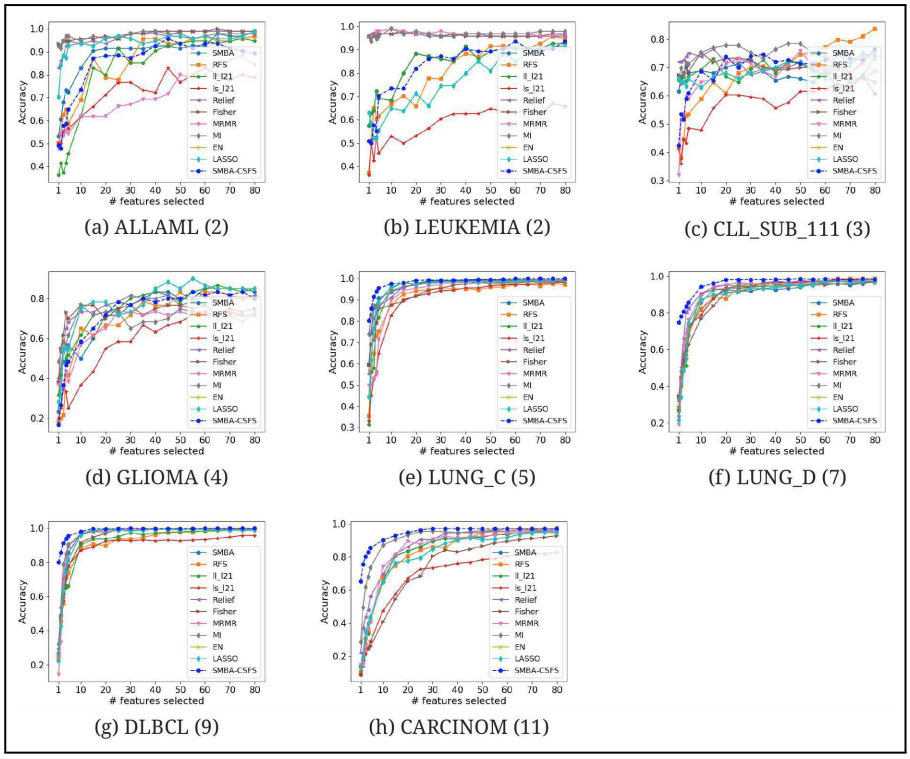

Figure 2. Comparison of several TFS accuracies against SMBA and SMBA-CSFS on eight datasets, when a varying number of features is selected. SVM classifier with 5-fold CV was used.

LUNG_D, CAR, DLBLC data sets in both cases, when top 20 and 80 features are considered. In this latter case, when SMBA-CSFS performs worse then its competitors, the corresponding performance tend to be comparable. On the remaining data sets, each with a number of classes less than 5, namely, ALLAML, LEUKEMIA, CLL_SUB_111 and GLIOMA, SMBA-CSFS is instead outperformed by some of the competitors. Consequently, we can assert that SMBA-CSFS behaves better when working with datasets with many classes (at least 5). One possible reason is due to the sparse-modeling approach in selecting the features and the use of an ensemble classifier. Indeed, since the ensemble is based on a majority voting schema, SMBA-CSFS is able to guess, with higher probability, the belonging of samples coming from data sets with many classes. Just think that, whenever our method draws from a sample of a two-class data set, the probability of a right guess is proportional to a coin toss. Therefore if, on one hand, this leads to good performance when the data set consists of of many classes, the probability of failure, on the other hand, increases in the case of data sets consisting of fewer classes. Anyhow, the local structure of data distribution which is crucial for feature selection, as stated in (He et al., 2005), may be a logical reason why the SBMA schema performs better on certain data set rather than others. In addition, as shown in Fig. 2, it is worth observing that SMBA-CSFS seems perform better w.r.t. TFS competitors on a fewer number of features. This would suggest that SMBA-CSFS is able to identify/retrieve the most representative features that maximize the classification accuracy. Concerning with the GF-CSFS competitors, looking at Fig. 3, it would suggest that the *sparse modeling* process, underlying the proposed SMBA scheme for feature selection, is more suitable for retrieving the best features for the purpose of classification w.r.t. the GF-CSFS, often leading to get satisfactory results. To statistically validate the results and compare all the competing classifiers against the proposed SMBA-CSFS, on both 20 and 80 feature subsets, we ran *Non-Parametric multiple comparison tests (all vs all)* (Demšar, 2006; Rodríguez-Fdez et al., 2015) which sequentially performs a popular multi-class *Friedman nonparametric test* (Friedman, 1937) followed by a Nemenyi *Post-hoc multiple comparison* (Dunn, 1961). The ranking of the classifiers, when the top 20 and 80 features are selected, along with the corresponding p-values, are described in the supplementary material. Looking at the *Cumulative Rank* (CR) for each classifier, one notices how SMBA-CSFS achieves optimal results (e.g., always ranks within the first three places).

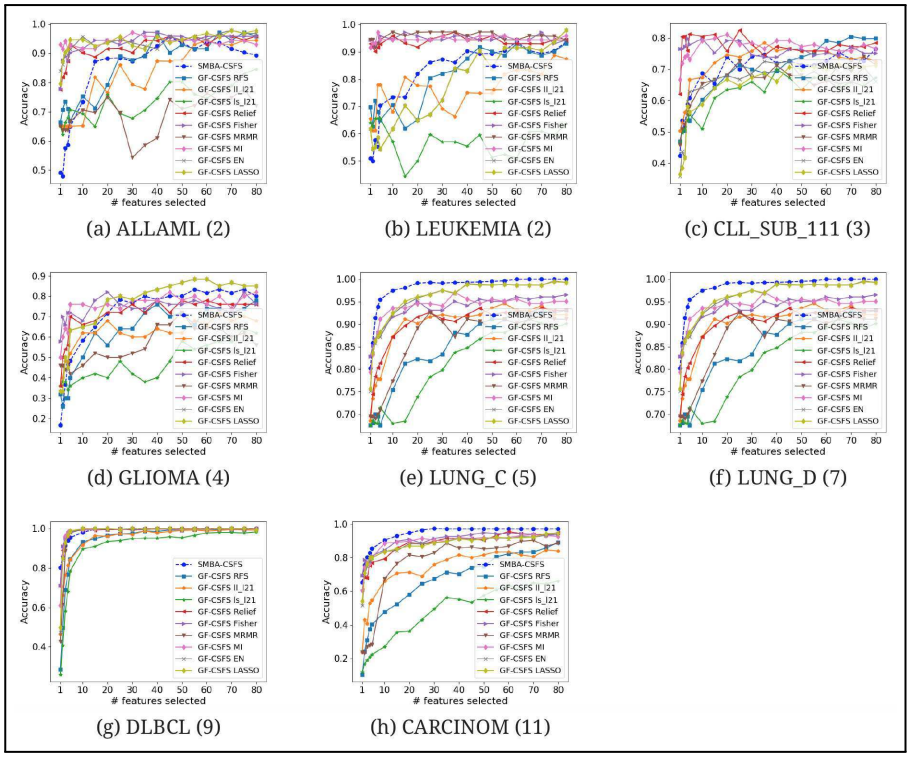

Figure 3. Comparison of several CSFS accuracies against SMBA-CSFS on eight datasets, when a varying number of features is selected. SVM classifier with 5-fold CV was used.

However, it is worth emphasizing that our method ranks systematically on the top place when considering datasets consisting of five or more classes (named $CR_{\geq 5}$). These results prove again that SMBA-CSFS has good performance on data sets with many classes. Moreover, by using different classifiers we do not observe noteworthy differences in the results, meaning that the methodology is suitable for the classification of this kind of data, independently from the selected classifier. However, by looking at the p-values, corresponding to the single ranking method, one can better verify which algorithms have significantly different performance w.r.t. SMBA-CSFS. Concerning the computational complexity, from several conducted experiments we observed that the proposed methodology might be slower than other techniques (e.g., FS and Relief whose running times are in term of few seconds) but comparable with SMBA. Its running time, depending on several parameters involved, especially in the size of the number of instances and classes of the datasets, might vary from a couple of hours to at most one day (see Table S9, in the Supplementary material, for details on the computational time). Nevertheless, SMBA-CSFS has appreciable performance when working on large datasets and number of classes, and sometimes, in the biological field, the accuracy in finding key features that are responsible for some biological processes would be preferred to the execution time. However, since most of the time consumed by the proposed approach is due to the solution of the optimization problem by using the ADMM method, and because the methodology is based on an ensemble of classifiers, a parallel computing approach could be adopted to obtain a faster computational time (Deng et al., 2017).

CONCLUSIONS

We proposed a Sparse-Modeling Based Approach for Feature Selection with emphasizing joint $\ell_{1,2}$-norm minimization and the Class-Specific Feature Selection. Experimental results, on eight different datasets, validate the unique aspects of SMBA-CSFS and demonstrate the promising performance achieved against the-state-of-art methods. One of the main characteristics of our framework is that, by jointly exploiting the idea of Sparse Modeling and Class-Specific Feature Selection, it is able to identify/retrieve the most representative features that maximize the classification accuracy in those cases where a given dataset is made up of many classes. Based on our experimental results, we can conclude that, usually applying TFS

allows achieving better results than using all the available features. However, in many cases, applying the proposed SMBA-CSFS method allows improving the performance of just TFS as well as GF-CSFS injected with several TFS methods. It has to be stressed, that SMBA-CSFS seems actually suitable for large datasets consisting of many classes, while on datasets with less than five classes other methods appear to be more effective. Although SMBA, SMBA-CSFS and TFS performance slightly differ on the whole, it is worth highlighting that SMBA-CSFS achieves its best performance when considering fewer features (i.e., from 1 to 20) on datasets with many classes, which is an important goal when certain biological tasks are taken into account. However, we do believe that these techniques might be effectively used in a systematic way after a microarray analysis. Indeed, a better gene selection step could avoid the waste of many resources in post-array wet analysis (e.g., Real Time-PCR) allowing researchers to focus their attention just on relevant features. Finally, we think this method demonstrated to be an interesting alternative among FS approaches on microarray data.

As future work, the focus will be moved towards the biologic interpretations of the SMBA framework behavior, by systematically studying the selected genes, especially taking into account the SMBA-CSFS approach which, as proved by the experimental results, is more effective in selecting genes of interest than the standard SMBA. Furthermore, we are planning to test our approach on EPIC dataset (Demetriou et al., 2013), after a thorough analysis of pre-filtering, and a parallel implementation to substantially reduce its computational time.

AVAILABILITY OF DATA AND MATERIALS

The data supporting the experiments in this article are available at the following data repository. For detailed information regarding the results, see the Supplementary material. A Python software package is available through GitHub repository containing all the source codes used to run SMBA-CSFS.

1 ACKNOWLEDGEMENTS

The research was entirely developed when Davide Nardone was a Master Degree student in Applied Computer Science at University of Naples Parthenope.

REFERENCES

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., et al. (2001). Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–13795.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122.

Calcagno, G., Staiano, A., Fortunato, G., Brescia-Morra, V., Salvatore, E., Liguori, R., Capone, S., Filla, A., Longo, G., and Sacchetti, L. (2010). A multilayer perceptron neural network-based approach for the identification of responsiveness to interferon therapy in multiple sclerosis patients. *Information Sciences*, 180(21):4153–4163.

Camastra, F., Di Taranto, M., and Staiano, A. (2015). Statistical and computational methods for genetic diseases: An overview. *Computational and Mathematical Methods in Medicine*, 2015(Article ID 954598).

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Ciaramella, A., Cocozza, S., Iorio, F., Miele, G., Napolitano, F., Pinelli, M., Raiconi, G., and Tagliaferri, R. (2008). Interactive data analysis and clustering of genomic data. *Neural Networks*, 21(2-3):368–378.

Ciaramella, A., Gianfico, M., and Giunta, G. (2016). Compressive sampling and adaptive dictionary learning for the packet loss recovery in audio multimedia streaming. *Multimedia Tools and Applications*, 75(24):17375–17392.

Ciaramella, A. and Giunta, G. (2016). Packet loss recovery in audio multimedia streaming by using compressive sensing. *IET Communications*, 10(4):387–392.

Demetriou, C. A., Chen, J., Polidoro, S., Van Veldhoven, K., Cuenin, C., Campanella, G., Brennan, K., Clavel-Chapelon, F., Dossus, L., Kvaskoff, M., et al. (2013). Methylome analysis and epigenetic changes associated with menarcheal age. *PloS one*, 8(11):e79391.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Deng, W., Lai, M.-J., Peng, Z., and Yin, W. (2017). Parallel multi-block admm with o (1/k) convergence. *Journal of Scientific Computing*, 71(2):712–736.

Di Taranto, M. D., Staiano, A., D'Agostino, M. N., D'Angelo, A., Bloise, E., Morgante, A., Marotta, G., Gentile, M., Rubba, P., and Fortunato, G. (2015). Association of usf1 and apoa5 polymorphisms with familial combined hyperlipidemia in an italian population. *Molecular and cellular probes*, 29(1):19–24.

Draghici, S., Khatri, P., Eklund, A., and Szallasi, Z. (2006). Reliability and reproducibility issues in dna microarray measurements. *Trends Genet.*, 22(2):101–109.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64.

Elhamifar, E., Sapiro, G., and Vidal, R. (2012). See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1600–1607. IEEE.

Engan, K., Aase, S. O., and Husoy, J. H. (1999). Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446. IEEE.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer, New-York.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Fu, X. and Wang, L. (2002). A ga-based rbf classifier with class-dependent features. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1890–1894. IEEE.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: class discovery and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27740v1 | CC BY 4.0 Open Access | rec: 17 May 2019, publ: 17 May 2019

class prediction by gene expression monitoring. *science*, 286(5439):531–537.

Gu, Q., Li, Z., and Han, J. (2012). Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3a725*.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Haslinger, C., Schweifer, N., Stilgenbauer, S., Döhner, H., Lichter, P., Kraut, N., Stratowa, C., and Abseher, R. (2004). Microarray gene expression profiling of b-cell chronic lymphocytic leukemia subgroups defined by genomic aberrations and vh mutation status. *Journal of Clinical Oncology*, 22(19):3937–3949.

He, X., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection, advances in nerual information processing systems.

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, New York, NY.

Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, pages 1200–1205. IEEE.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer.

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.

Kreyszig, E. (2010). *Advanced engineering mathematics*. John Wiley & Sons, Great Britain.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2009). Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE.

Nardone, D., Ciaramella, A., and Staiano, A. (2019a). Biological datasets. `https://zenodo.org/record/2709491#.XNV4AegzaUk`.

Nardone, D., Ciaramella, A., and Staiano, A. (2019b). Source code. `https://github.com/DavideNardone/A-Sparse-Coding-Based-Approach-for-Class-Specific-Feature-Sele`

Nie, F., Huang, H., Cai, X., and Ding, C. H. (2010). Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821.

Nutt, C. L., Mani, D., Betensky, R. A., Tamayo, P., Cairncross, J. G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M. E., Batchelor, T. T., et al. (2003). Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer research*, 63(7):1602–1607.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.

Pineda-Bautista, B. B., Carrasco-Ochoa, J. A., and Martınez-Trinidad, J. F. (2011). General framework for class-specific feature selection. *Expert Systems with Applications*, 38(8):10018–10024.

Ramirez, I., Sprechmann, P., and Sapiro, G. (2010). Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE.

Rodríguez-Fdez, I., Canosa, A., Mucientes, M., and Bugarín, A. (2015). Stac: a web platform for the comparison of algorithms using statistical tests. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–8. IEEE.

Ross, B. C. (2014). Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Staiano, A., De Vinco, L., Ciaramella, A., Raiconi, G., Taglaiferri, R., Amato, R., and et al. (2004). Probabilistic principal surfaces for yeast gene microarray data mining. In *Proceedings of the Fourth*

IEEE International Conference on Data Mining, ICDM 2004, pages 202–208. IEEE.

Staiano, A., Di Taranto, M. D., Bloise, E., D'Agostino, M. N., D'Angelo, A., Marotta, G., Gentile, M., Jossa, F., Iannuzzi, A., Rubba, P., et al. (2013). Investigation of single nucleotide polymorphisms associated to familial combined hyperlipidemia with random forests. In *Neural Nets and Surroundings*, volume 19, pages 169–178. Springer, Berlin, Heidelberg.

Su, A. I., Welsh, J. B., Sapinoso, L. M., Kern, S. G., Dimitrov, P., Lapp, H., Schultz, P. G., Powell, S. M., Moskaluk, C. A., Frierson, H. F., et al. (2001). Molecular classification of human carcinomas by use of gene expression signatures. *Cancer research*, 61(20):7388–7393.

Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, page 37.

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Xiong, M., Fang, X., and Zhao, J. (2001). Biomarker identification by feature wrappers. *Genome Research*, 11(11):1878–1887.

Yang, K., Cai, Z., Li, J., and Lin, G. (2006). A stable gene selection in microarray data analysis. *BMC bioinformatics*, 7(1):228.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Table 1. Datasets Description.

	Size	# of Features	# of Classes
ALLAML	72	7129	2
LEUKEMIA	72	7070	2
CLL_SUB_111	111	11340	3
GLIOMA	50	4434	4
LUNG_C	203	3312	5
LUNG_D	73	325	7
DLBCL	96	4026	9
CAR	174	9182	11

Table 2. SVM accuracy results (ACC±STD) on top 20 features using 5-fold CV on different datasets.

TFS methods are compared against our methods (SMBA and SMBA-CSFS). FS: Fisher Score, mRmR: Minimum-Redundancy-Maximum-Relevance, MI: Mutual Information, RFS: Robust Feature Selector, EN: Elastic Net, BSL: all features. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

Average Accuracy of top 20 features (%)								
	ALLAML	LEUKEMIA	CLL_SUB_111	GLIOMA	LUNG_C	LUNG_D	DLBCL	CAR
Fisher	96.84±0.04(19)	98.95±0.02(16)	75.20±0.1(19)	80±0.04(13)	91.94±0.02(19)	91.24±0.1(20)	97.11±0.02(19)	65.33±0.05(20)
Relief	95.78±0.04(8)	97.89±0.03(12)	76.45±0.03(15)	80±0.07(19)	97.12±0.01(20)	95.2±0.03(14)	99.76±0.00(20)	86.52±0.03(18)
mRmR	66.14±0.13(12)	**98.95±0.02(9)**	71.27±0.1(20)	66.67±0.1(17)	95.68±0.013(19)	95.22±0.02(20)	99.03±0.01(16)	89.57±0.04(20)
MI	96.84±0.042(15)	98.95±0.02(10)	**81.03±0.06(17)**	78.33±0.04(12)	97.41±0.014(17)	94.53±0.03(18)	98.79±0.01(19)	93.25±0.05(20)
ls-21	71.34±0.14(19)	59.42±0.2(12)	60.30±0.14(19)	55±0.07(20)	92.66±0.05(19)	93.86±0.04(20)	92.52±0.01(20)	66.99±0.03(20)
ll-21	83±0.11(15)	88.36±0.06(20)	73.12±0.06(15)	0.75±0.12(17)	98.27±0.015(16)	93.24±0.04(16)	94.44±0.02(19)	83.49±0.03(20)
RFS	87±0.01(15)	74.33±0.1(18)	64.73±0.09(15)	66.67±0.07(17)	94.10±0.022(20)	89.77±0.02(19)	91.06±0.03(18)	81.85±0.07(18)
LASSO	**98.95±0.02(17)**	71.3±0.08(21)	68.02±0.06(20)	**83.33±0.05(17)**	97.99±0.012(16)	92.51±0.03(12)	**99.52±0.01(16)**	82.14±0.05(18)
EN	**98.95±0.02(17)**	71.3±0.08(21)	68.02±0.06(20)	**83.33±0.05(17)**	97.99±0.012(16)	92.51±0.03(12)	**99.52±0.01(16)**	82.14±0.05(18)
SMBA	93.68±0.084(16)	88.36±0.06(20)	70.60±0.10(19)	71.67±0.134(17)	97.84±0.00(20)	92.55±0.03(20)	99.28±0.01(20)	83.49±0.03(20)
SMBA-CSFS	88.24±0.04(20)	81.93±0.02(20)	75.53±0.06(20)	73.34±0.18(16)	**98.41±0.014(19)**	**97.93±0.03(19)**	98.30±0.02(13)	**94.95±0.02(19)**
BSL	97.89±0.04	98.95±0.021	84.26±0.06	85±0.1	99.57±0.00	98.62±0.02	100±0.00	98.65±0.01

Table 3. SVM accuracy results (ACC±STD) on top 80 features using 5-fold CV on different datasets.

TFS methods are compared against our methods (SMBA and SMBA-CSFS). FS: Fisher Score, mRmR: Minimum-Redundancy-Maximum-Relevance, MI: Mutual Information, RFS: Robust Feature Selector, EN: Elastic Net, BSL: all features. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

Average Accuracy of top 80 features (%)								
	ALLAML	LEUKEMIA	CLL_SUB_111	GLIOMA	LUNG_C	LUNG_D	DLBCL	CAR
Fisher	**99.95±0.00(65)**	98.95±0.02(16)	75.87±0.06(80)	80±0.04(13)	99±0.00(79)	96.6±0.02(69)	99.76±0.00(27)	92.92±0.02(72)
Relief	98.94±0.02(38)	98±0.03(12)	76.45±0.03(15)	83.33±0.12(58)	99.57±0.00(77)	97.29±0.014(46)	99.76±0.00(20)	96.64±0.01(80)
mRmR	88.30±0.05(75)	98.95±0.02(9)	75.85±0.13(50)	75±0.07(29)	99.14±0.01(44)	97.95±0.02(74)	**100±0.00(77)**	95.61±0.02(75)
MI	98.94±0.02(31)	**99±0.02(10)**	81.03±0.06(17)	78.33±0.04(12)	99.42±0.01(60)	97.95±0.02(73)	99.52±0.01(31)	97.31±0.01(65)
ls-21	83.1±0.01(60)	67.89±0.18(73)	68.67±0.05(79)	76.67±0.06(61)	98.42±0.02(78)	97.95±0.02(65)	96.12±0.03(73)	82.8±0.04(80)
ll-21	96.84±0.04(76)	93.68±0.04(73)	73.94±0.07(49)	86.67±0.07(61)	99.28±0.01(43)	97.26±0.03(61)	98.79±0.01(63)	95.97±0.01(78)
RFS	97.9±0.03(76)	95.8±0.05(72)	**83.61±0.07(79)**	85±0.06(66)	97.84±0.01(62)	**98.62±0.02(65)**	99.51±0.01(72)	97.31±0.01(80)
LASSO	98.94±0.02(17)	92.51±0.04(74)	75.27±0.08(75)	**91.67±0.09(57)**	99.42±0.01(79)	97.29±0.01(58)	99.76±0.00(59)	95.28±0.02(73)
EN	98.94±0.02(17)	92.51±0.04(74)	75.27±0.08(75)	**91.67±0.09(57)**	99.42±0.01(79)	97.29±0.01(58)	99.76±0.00(59)	95.28±0.02(73)
SMBA	98.94±0.02(78)	93.68±0.04(73)	75.91±0.13(27)	88.33±0.04(66)	**99.71±0.01(45)**	97.26±0.01(79)	99.76±0.00(29)	95.97±0.01(78)
SMBA-CSFS	95.79±0.04(43)	95.73±0.04(77)	77.18±0.08(79)	83.33±0.11(28)	99.42±0.01(27)	**98.62±0.03(27)**	98.54±0.02(22)	**98.65±0.013(56)**
BSL	97.89±0.04	98.95±0.021	84.26±0.06	85±0.1	99.57±0.00	98.62±0.02	100±0.00	98.65±0.01

Table 4. SVM accuracy results (ACC±STD) on top 20 features using 5-fold CV on different datasets.

GF-CSFS $(Pineda-Bautistaet\ al., 2011)$ framework is compared against our SMBA-CSFS. FS: Fisher Score, mRmR: Minimum-Redundancy-Maximum-Relevance, MI: Mutual Information, RFS: Robust Feature Selector, EN: Elastic Net, BSL: all features. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

Average Accuracy of top 20 features (%)								
	ALLAML	LEUKEMIA	CLL_SUB_111	GLIOMA	LUNG_C	LUNG_D	DLBCL	CAR
Fisher	**95.90±0.03(13)**	**98.57±0.03(18)**	80.41±0.02(7)	**82±0.16(17)**	95.09±0.03(20)	86.38±0.14(16)	**100±0.00(14)**	90.86±0.08(20)
Relief	92.95±0.04(5)	95.81±0.03(10)	**82.41±0.05(12)**	80±0.19(12)	91.63±0.02(20)	86.39±0.07(20)	**100±0.00(11)**	89.68±0.03(17)
mRmR	75.14±0.09(16)	**98.57±0.03(11)**	70.69±0.07(12)	62±0.12(14)	89.16±0.03(20)	86.48±0.09(17)	99.52±0.01(15)	81.61±0.07(20)
MI	94.38±0.03(18)	97.14±0.03(4)	81.03±0.05(20)	**82±0.21(19)**	95.07±0.015(11)	79.90±0.18(14)	**100±0.00(19)**	90.86±0.06(11)
ls-21	76.47±0.13(6)	65.52±0.08(3)	63.44±0.03(20)	46±0.21(7)	73.88±0.04(19)	75.43±0.07(18)	93.46±0.03(20)	39.68±0.04(19)
ll-21	82.1±0.05(16)	80.67±0.09(15)	74.58±0.07(20)	68±0.13(18)	91.15±0.02(15)	67.24±0.12(15)	96.38±0.02(17)	72.40±0.05(17)
RFS	79.24±0.168(17)	74.95±0.09(6)	71.94±0.10(19)	68±0.21(13)	82.79±0.05(17)	68.67±0.07(18)	96.62±0.01(20)	58.03±0.18(20)
LASSO	95.73±0.02(6)	70.3±0.08(15)	71.29±0.05(18)	81.67±0.08(19)	96.26±0.00(18)	93.22±0.021(20)	**100±0.00(10)**	87.88±0.03(18)
EN	95.73±0.04(10)	70.3±0.08(15)	68.73±0.10(19)	81.67±0.08(19)	95.97±0.012(18)	93.22±0.021(20)	**100±0.00(10)**	88.56±0.03(19)
SMBA-CSFS	88.24±0.04(20)	81.93±0.02(20)	75.53±0.06(20)	73.34±0.18(16)	**98.41±0.014(19)**}	**97.93±0.03(19)**	98.30±0.02(13)	**94.95±0.02(19)**
BSL	97.89±0.04	98.95±0.021	84.26±0.06	85±0.1	99.57±0.00	98.62±0.02	100±0.00	98.65±0.01

Table 5. SVM accuracy results (ACC±STD) on top 80 features using 5-fold CV on different datasets.

GF-CSFS $(Pineda-Bautistaet\ al., 2011)$ framework is compared against SMBA-CSFS. FS: Fisher Score, mRmR: Minimum-Redundancy-Maximum-Relevance, MI: Mutual Information, RFS: Robust Feature Selector, EN: Elastic Net, BSL: all features. The best results are highlighted in bold. The number in parentheses is the number of features when the performance is achieved.

Average Accuracy of top 80 features (%)								
	ALLAML	LEUKEMIA	CLL_SUB_111	GLIOMA	LUNG_C	LUNG_D	DLBCL	CAR
Fisher	97.24±0.03(35)	98.57±0.03(18)	80.41±0.02(7)	84±0.17(33)	96.56±0.02(72)	86.38±0.14(16)	**100±0.00(14)**	94.86±0.05(56)
Relief	97.24±0.03(48)	**98.67±0.03(29)**	82.41±0.05(12)	82±0.13(49)	93.61±0.02(45)	86.48±0.07(71)	**100±0.00(11)**	95.43±0.05(60)
mRmR	80.47±.05(53)	**98.67±0.03(37)**	73.98±0.09(75)	72±0.16(50)	92.62±0.03(25)	86.48±0.09(17)	99.76±0.00(21)	90.82±0.07(71)
MI	97.14±0.04(30)	97.24±0.03(53)	81.7±0.03(21)	84±0.14(41)	97.05±0.02(35)	84.95±0.03(43)	**100±0.00(19)**	93.71±0.06(68)
ls-21	84.57±0.14(80)	65.52±0.08(3)	0.7±0.08(40)	68±0.17(79)	91.13±0.04(65)	85.14±0.1(72)	98.8±0.01(76)	67.26±0.04(78)
ll-21	95.81±0.03(67)	88.76±0.04(75)	78.45±0.09(35)	72±0.19(65)	94.59±0.02(55)	82.19±0.09(72)	99.52±0.01(57)	85.13±0.06(53)
RFS	97.24±0.03(65)	93.05±0.00(60)	**82.37±0.04(76)**	78±0.2(37)	93.59±0.04(66)	86.48±0.10(58)	99.76±0.00(58)	89.11±0.07(79)
LASSO	**97.9±0.04(57)**	97.89±0.03(80)	74.54±0.07(79)	**91.67±0.07(56)**	**99.57±0.01(74)**	96.55±0.03(72)	**100±0.00(10)**	94.94±0.02(73)
EN	**97.9±0.04(57)**	97.89±0.03(80)	74.47±0.04(41)	**91.67±0.07(56)**	**99.57±0.01(74)**	96.55±0.03(72)	**100±0.00(10)**	94.60±0.03(78)
SMBA-CSFS	95.79±0.04(43)	95.73±0.04(77)	77.18±0.08(79)	83.33±0.11(28)	99.42±0.01(27)	**98.62±0.03(27)**	98.54±0.02(22)	**98.65±0.013(56)**
BSL	97.89±0.04	98.95±0.021	84.26±0.06	85±0.1	99.57±0.00	98.62±0.02	100±0.00	98.65±0.01

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27740v1 | CC BY 4.0 Open Access | rec: 17 May 2019, publ: 17 May 2019