PLOS ONE

RESEARCH ARTICLE

Accurate and fast feature selection workflow for high-dimensional omics data

Yasset Perez-Riverol[1]*, Max Kuhn[2], Juan Antonio Vizcaíno[1], Marc-Phillip Hitz[3,4,5,6], Enrique Audain[3,4,5]*

1 European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge, United Kingdom, **2** RStudio Inc., Boston, MA, United States of America, **3** Department of Congenital Heart Disease and Pediatric Cardiology, Universitätsklinikum Schleswig–Holstein Kiel, Kiel, Germany, **4** German Center for Cardiovascular Research (DZHK), Berlin, Germany, **5** Department of Human Genetics, University Medical Center Schleswig-Holstein (UKSH), Kiel, Germany, **6** Wellcome Trust Sanger Institute, Cambridge, United Kingdom

* yperez@ebi.ac.uk (YPR); enrique.audain@uksh.de (EA)

Citation: Perez-Riverol Y, Kuhn M, Vizcaíno JA, Hitz M-P, Audain E (2017) Accurate and fast feature selection workflow for high-dimensional omics data. PLoS ONE 12(12): e0189875. https://doi.org/10.1371/journal.pone.0189875

Editor: Quan Zou, Tianjin University, CHINA

Received: June 27, 2017

Accepted: December 4, 2017

Published: December 20, 2017

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: YP-R is supported by BBSRC 'PROCESS' grant (BB/K01997X/1). JAV acknowledges the Wellcome Trust (grant number WT101477MA) and EMBL core funding. EA and MPH are supported by DZHK (German Center for Cardiovascular Research), partner sites: Kiel, Germany. The funder provided support in the form of salaries for authors [MK], but did not have any additional role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. The

Abstract

We are moving into the age of 'Big Data' in biomedical research and bioinformatics. This trend could be encapsulated in this simple formula: $D = S * F$, where the volume of data generated (D) increases in both dimensions: the number of samples (S) and the number of sample features (F). Frequently, a typical omics classification includes redundant and irrelevant features (e.g. genes or proteins) that can result in long computation times; decrease of the model performance and the selection of suboptimal features (genes and proteins) after the classification/regression step. Multiple algorithms and reviews has been published to describe all the existing methods for feature selection, their strengths and weakness. However, the selection of the correct FS algorithm and strategy constitutes an enormous challenge. Despite the number and diversity of algorithms available, the proper choice of an approach for facing a specific problem often falls in a 'grey zone'. In this study, we select a subset of FS methods to develop an efficient workflow and an R package for bioinformatics machine learning problems. We cover relevant issues concerning FS, ranging from domain's problems to algorithm solutions and computational tools. Finally, we use seven different proteomics and gene expression datasets to evaluate the workflow and guide the FS process.

Introduction

The term 'Big Data' is often used to describe the huge volumes of information produced by modern systems such as mobile devices, tracking tools and sensors [1, 2]. In biomedical research, the growth of high-throughput (omics) technologies has resulted in an exponential growth in the dimensionality and sample size. This increase has two major directions: i) the number of samples processed, powered by novels machines (i.e. sequencers and mass spectrometers); and ii) the features, attributes and variables collected alongside each sample [3]. This high-dimensional environment becomes a challenge to many modelling tasks used in bioinformatics, ranging from sequence analysis to spectral analyses as well as literature mining.

specific roles of these authors are articulated in the 'author contributions' section.

Competing interests: The authors have declared that no competing interests exist. Even though one of the authors (MK) is affiliated to a commercial company (RStudio Inc.), this does not alter our adherence to PLOS ONE policies on sharing data and materials.

Abbreviations: CM, Correlation Matrix; FS, Feature Selection; ML, Machine Learning; PCA, Principal Component Analysis; RFE, Recursive Feature Elimination; RMSE, Root Mean Square Error; RF, Random Forest; SVM, Support Vector Machine; TNBC, Triple-Negative Breast Cancer; X2, Univariate Correlation.

Reducing data complexity is therefore crucial for data analysis tasks, knowledge inference using machine learning (ML) algorithms, and data visualization [4–6].

The 'curse of dimensionality' (term first introduced by Bellman in 1957) [7] described the problem caused by the exponential increase in volume associated with adding extra dimensions to an Euclidean space. In this context, the typical bioinformatics problem involves both: relevant and redundant features. Therefore, a Feature Selection (FS) approach becomes a crucial and non-trivial task because: i) it provides a deeper insight into the underlying processes that are the foundation of the data; ii) it improves the performance (CPU-time and memory) of the ML step, by reducing the number of variables; and iii) it produces better model results avoiding overfitting. However, a FS algorithm brings an important decision in any ML workflow (e.g. classification of protein/gene expression profiles): are there redundant features (e.g. proteins or genes) in the dataset that are irrelevant and/or redundant for the biological study?

The most-common attempt to address the FS problem (the so-called univariate filtering approach) is to use a variable ranking method to filter out the least promising variables before using a multivariate method [8]. These methods have been used extensively in computational biology for cancer classification using microarray data [9, 10]. However, correlation filters could prompt some loss of relevant features that are meaningless by themselves but that can be useful in combination. To overcome this effect, a set of algorithms has been proposed to combine the original variables into a new and smaller subset of features, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis. In PCA [11], new orthogonal features (latent variables or principal components) are obtained by maximizing the variation of the original features. The number of the latent features (factors) can be much lower than the number of original features, so that the data can be visualized in a much lower-dimensional space. As correlation filters, PCA methods can reduce the number of variables by looking into the feature dependencies without taking into account the final learning model. In 1997, a powerful strategy emerged that combines a FS algorithm with a learning/classification step: the so-called *wrapper* methods [12]. These wrapper approaches (e.g. *forward selection* and *backward elimination*) can use the prediction performance of a given ML approach to assess the relative usefulness of different subsets of variables. An exhaustive search can be performed if the number of variables is not too large.

Due to the diversity of FS methods available, it is hard to choose the correct approach needed to accomplish a specific task beforehand (e.g. regression or classification). In 2007, Saeys and co-workers published an introduction to FS in bioinformatics [3]. Also, several reviews have focused on the application in computational biology of particular methods such as PCA [13, 14] or Support Vector Machines (SVM) [15]. However, most of this work has been done to describe current methods in isolation and not to evaluate how they could be combined. In this manuscript, we developed a FS workflow and an R package for high-dimensional omics data analysis. The workflow combined univariate/multivariate correlation filters with wrapper feature backward elimination and it was applied to regression and classification problems. We benchmarked the individual steps of the described workflow, highlighting the optimal steps in different scenarios, using seven different omics datasets. Finally, we discuss major challenges when applying the described workflow to classification problems of high-dimensional omics data.

Materials and methods

Transcriptomic dataset of breast tumor samples (Dataset 1)

We first used a gene expression dataset (GEO (Gene Expression Omnibus) accession number: GSE5325) from *Saal et al.* [16], which has already been extensively studied before [13]. The

authors performed a study using microarrays to measure the expression of 27,648 genes in 105 breast tumor samples. The dataset includes the estrogen receptor alpha status (0 = negative, 1 = positive), a transcription factor recognized as being important for stimulating the growth of a large proportion of breast cancers and used to explore co-expression [17].

High-resolution isoelectric focusing proteomics dataset (Dataset 2)

The second dataset is the result of an electrophoresis experiment on peptide samples [18]. A total of 7,391 peptides were identified in 12 fractions, where each fraction corresponded to an experimental isoelectric point. This dataset has been used before to develop a ML model that can accurately predict the theoretical isoelectric points for peptides and proteins based on the amino acid sequence properties [5, 19].

Triple-Negative Breast Cancer (TNBC) dataset (Dataset 3)

A third dataset containing protein quantification data using a label free technique was included [20]. The dataset assembles a panel of 44 (including samples and technical replicates) human breast cell lines and clinical tumors for analyzing the proteomics landscape of TNBC. The studied cell lines cover mesenchymal-, luminal-, and basal-like subtypes, as well as three receptor-positive and one non-tumorigenic cell lines. Thus, the idea behind including this dataset was to evaluate the ability of the proposed FS workflow to classify subtypes of cellular lines.

Transcriptomics analysis of left ventricles of mouse hearts (Dataset 4)

A fourth dataset included the results of a transcriptomics analysis of left ventricles of mouse hearts subjected to an isoproterenol challenge [21]. In the study, the authors utilized expression arrays from left ventricular (LV) tissues, with and without an isoproterenol treatment, to understand the genetic control of gene expression and its relationship with heart failure. Then, the issue arising here suggests a binary classification problem where the researcher could be interested in, in order to know the optimal feature subset which could best discriminate between both classes (treated and non-treated samples).

Expression data from normal and prostate tumor tissues (Datasets 5, 6, and 7)

Recently, Li *et al*. have used several gene expression datasets to benchmark different FS algorithms [22]. From the original microarray datasets, we have selected three of those datasets (GEO accession number: GSE6919), to compare the FS workflow with the results obtained by *Li et al*. **Note 1** (**S1 File**) summarizes the main characteristics of the datasets described previously.

Workflow R-package

An R-package has been developed to reproduce the proposed workflow (https://github.com/enriquea/feseR). For its development five main R packages were used: i) **Caret** [23] (**C**lassification **A**nd **RE**gression **T**raining) (http://topepo.github.io/caret), containing a set of functions that attempt to streamline the process for creating predictive models; ii) **randomForest** [24], a package enabling Random Forest analysis (https://cran.r-project.org/web/packages/randomForest/); iii) **prcomp**, a native function included in the R package *stats*; iv) ***Kernlab*** [25] (https://cran.r-project.org/web/packages/kernlab/), which provides the user with basic kernel functionality (e.g., computing a kernel matrix), along with some utility functions,

commonly used in kernel-based methods; and v) the **FSelector** package [26] (https://cran.r-project.org/web/packages/FSelector/), which offers algorithms for filtering attributes (e.g. chi-squared, information gain, and linear correlation).

We have used the current FS workflow and R-package in combination for two different ML (regression/classification) problems. Six of the datasets represent classification of (protein/gene) expression profiles and the last one a regression problem for the accurate estimation of the isoelectric point of peptides and proteins. In the following sections, we discuss the results of combining the different steps of the FS workflow depending of the ML problem.

Results and discussion

A good feature subset can be defined as one that contains features highly correlated with (predictive of) outcome, yet uncorrelated (independent) with (not predictive of) each other. Nevertheless, the existing diversity of FS methods makes it challenging to choose the correct one for the task at hand (**S1 File, Note 2**). **Fig 1** represents the proposed overall workflow to perform FS in high-dimensional omics big data. First, a univariate correlation filter can be used before applying any wrapper approach, to determine the relation between each feature and the class or predicted variable. Then, a second filtering step (Correlation Matrix (CM) or PCA), can follow, in order to determine the dependencies between the different dataset features. Finally, backward elimination is achieved by wrapping a ML method, such as Random Forest and SVM around each example.

Removing irrelevant features: Univariate correlation filtering

The univariate correlation filtering step removes all features that are not directly related to their class variables. When we applied this approach to **Dataset 1** it removed those genes with a non-correlated expression to the presence or absence of estrogen receptor alpha, reducing the number of genes from 8,534 (only those genes showing expression in all samples were considered) to 1,697. In **Dataset 2**, we used the univariate filter to remove features (amino acid properties) unrelated with the isoelectric point. **Fig 2A** shows the high-correlation found among the original 545 physicochemical peptides properties considered for the 7,391 peptides. We implemented a univariate correlation filter to remove all features that were not correlated with the isoelectric point (correlation coefficient $< = 0.30$), reducing the number of variables to 89 features. When we extended the analysis to the remaining benchmarking datasets, we observed that, in general, univariate correlation filtering removed more than 80% of the original features that were not related to the predicted variable. As previously discussed by other

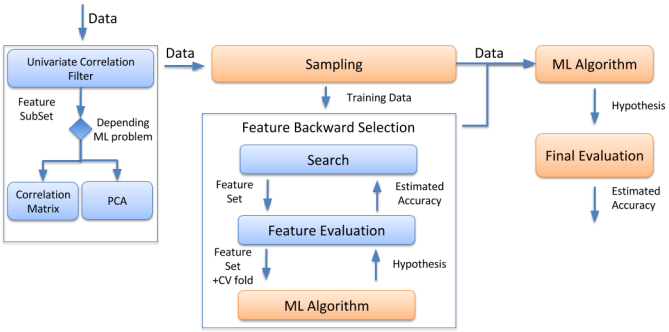

Fig 1. Proposed workflow for FS including a filtering step with univariate and/or multivariate approaches, followed by a wrapper approach (recursive feature elimination).

https://doi.org/10.1371/journal.pone.0189875.g001

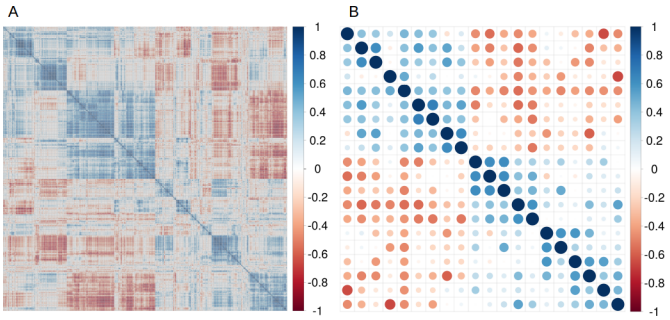

Fig 2. (A) Correlation matrix for the 544 physicochemical (features) of the 7,391 peptides (samples) included in Dataset 2; (B) the final 20 variables after the correlation-matrix filtering steps.

https://doi.org/10.1371/journal.pone.0189875.g002

authors [8], univariate correlation filtering should be always applied at early stages of any classification and/or regression process. However, univariate correlation filtering can only be used to study the relationship of each feature with a class variable, but cannot be applied to find the relationships among them. For this reason, a multivariate step (e.g. correlation matrix) was used (**Fig 1**) to remove the redundancy among highly-correlated features (correlation coefficient $>= 0.75$).

Reducing feature complexity: CM or PCA

We implemented two different strategies (depending on the classification or regression problem) to reduce the number of variables, while keeping most of the original and relevant information: CM and PCA. **Dataset 2** is a good example of a dataset containing regression related problems. In this particular case, the aim was to predict more accurately the isoelectric point of peptides and proteins, using other physicochemical features of the peptides. Therefore, the final model should be based on, or be correlated to, the original features (because they would be used in the future to make a predictor that could be applied for other datasets). One of the simplest and most powerful filtering approaches to remove feature redundancy, while keeping original features, is the use of a CM filter. For example, peptides properties such as aromatic rings, bond and carbon atom counts are strongly correlated [5, 27]. Therefore, any of these variables could be used as a proxy for all the others. It should be noted that several features clustered together, suggesting a high-redundancy in the feature set. By applying the CM filter, it is possible to remove those that are redundant (or irrelevant) and to keep only a reduced feature set for subsequent analysis steps. The present workflow keeps only 20 variables (out of the original 545 features, see **Fig 2B**) for the final ML step (**Fig 1**). The current approach also reuses the final model in new datasets because the filtering steps preserve the original variables by only removing the redundant ones.

Opposite to **Dataset 2**, the other datasets constitute good examples of classification related problems. In addition to the CM filter approach, we implemented and studied the use of Principal Component Analysis (PCA) as a multivariate filter to reduce the number of features. PCA reduces the dimensionality of the data while retaining most of the variation in the predictor variables [13]. Thus, by using a few components, PCA can represent each sample by using relatively few (new) variables instead of (potentially) thousands of them. **Fig 3** shows the PCA performed in **Dataset 1**. The proportion of the variation present in all genes is encompassed within each of the principal components, with the first few components representing most of it (**Fig 3A**). The cumulative variance analysis shows that most of the variance is contained in the first 30 principal components (75%), where only 76 components reach a 95% of variance

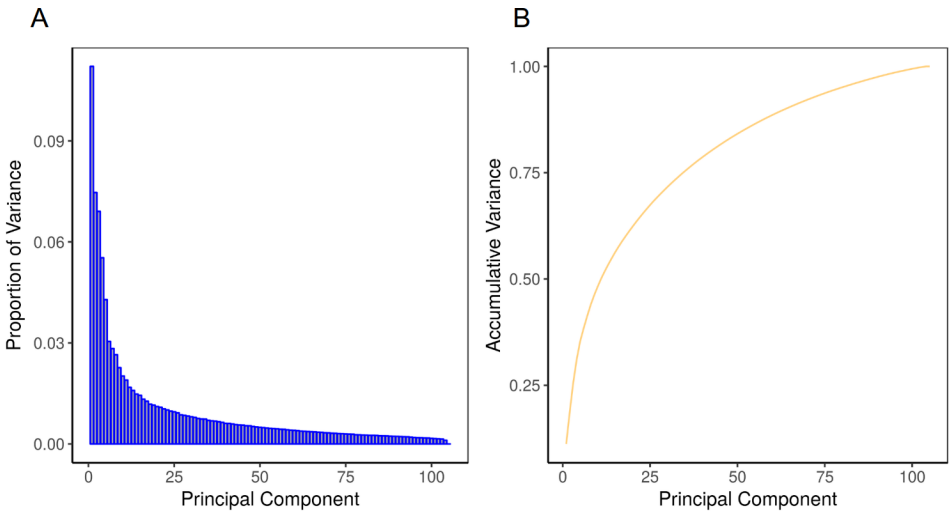

Fig 3. (A) Proportion of variance and (B) cumulative variance of principal components for the analysis of Dataset 1.

https://doi.org/10.1371/journal.pone.0189875.g003

(Fig 3B) and 104 components are enough to retain all the original variance. This number of variables is 10-fold smaller than the original 1,697 features obtained after applying the univariate correlation filter.

When the number of variables is larger than the number of samples, PCA can reduce the dimensionality of the samples to, at most, the number of samples, without losing information [13, 28]. We obtained the same results when PCA was applied to the other relevant datasets (**Dataset 3** to **Dataset 7**, those with a classification problem, S2 File). However, since the principal components are linear combinations of the original data, it is not obvious how model parameter estimates can relate back to the original variables. Thus, this method is not suitable for problems where it is required to keep the primary information (e.g. in the case of regression problems, **Dataset 2**).

Optimizing the feature selection: Wrapper recursive feature elimination

All filtering FS approaches previously shown (e.g. correlation-based or PCA) are relatively easy to implement and computationally fast. Therefore, these algorithms represent a suitable choice in the first stage of any given FS pipeline. However, wrapper methods should be used in the last steps to find the "optimal" feature subsets, by iteratively selecting features based on classifier performance (Fig 1). The wrapper methods should be combined with cross-validation steps to improve the final results [12, 29]. These cross-validation steps can be used to assess the results of the learning analysis (e.g. regression or classification) and help to generalize these steps to an independent dataset. The goal of cross-validation is to define a dataset to "test" the model in the training phase (i.e., the validation dataset), in order to limit problems like overfitting [29]. In the proposed workflow, we used a recursive feature elimination (backward elimination) approach in combination with two ML models (Random Forest and SVM) to systematically increase each ML step. The number of cross-validation iterations should be evaluated in detail because it could significantly increase the running time without improving the performance of the model prediction.

We implemented the wrapper backward elimination step in combination with the SVM radial kernel, in order to predict the isoelectric point using **Dataset 2**. Table 1 shows the performance (regarding running time and model prediction accuracy) of the feature workflow for

Table 1. Benchmark of the SVM regression model for Dataset 2 applying different FS methods (SVM), no feature selection, (X2) univariate correlation alone, (CM) correlation matrix filtering, (RFE) and wrapper feature elimination. The figures indicated using the prefixes CV3, CV7 and CV10 correspond to the number of interactions in the cross-validation steps during the RFE feature selection.

	R^2	RMSE	Time (min)	# Features
SVM	0.97	0.88	6.8	545
X2-CM-SVM	0.98	0.57	0.5	28
RFE-SVM-CV3	0.98	0.32	35	4
RFE-SVM-CV7	0.98	0.32	115	4
RFE-SVM-CV10	0.98	0.32	168	4
X2-CM-RFE-SVM-CV3	0.98	0.33	11	2
X2-CM-RFE-SVM-CV7	0.98	0.34	36	2
X2-CM-RFE-SVM-CV10	0.98	0.34	48.1	2

https://doi.org/10.1371/journal.pone.0189875.t001

Dataset 2. We benchmarked all the FS combinations with the SVM model by removing each of them. Applying the SVM model alone (**SVM**) without FS or cross-validation helps to predict the isoelectric point with a high root-mean-square error (RMSE) of 0.88. In contrast, when both correlation filters (**X2-CM-SVM**) were applied, RMSE and running time decreased to 0.57 and 0.50 min, respectively. When the complete workflow (**X2-CM-RFE-SVM-CV3**) was used RMSE decreased to 0.33 (Table 1). It should be noted that when pre-filtering was applied (**RFE-SVM-CV3**), RMSE decreased to 0.32 and two new variables were added to the SVM model. However, this improvement in performance (e.g. low RMSE) decreased the overall efficiency of the workflow by increasing the execution time three-fold. Also, we observed no changes where the number of cross-validation steps was increased (Table 1).

Wrapper backward elimination step provided a powerful method to optimize the final subset of variables in response to the regression SVM model. Fig 4 shows the final results of the isoelectric point prediction (**Dataset 2**) for all FS combinations. Backward selection in combination with the cross-validation step enables a better estimation of the variable prediction (isoelectric point) in the regions where less experimental evidences exist (basic pH range). This workflow has been used in a recent approach to predict the isoelectric point and it has proven to predict the isoelectric point more accurately than any other algorithm so far. A similar implementation was applied to the remaining datasets (1, 3–7) where a Random Forest model was wrapped around, using a recursive approach to evaluate the performance and the variable weight following different FS workflows. We first evaluated the Random Forest approach for FS without any filtering and parameter tuning as discussed before by *Díaz-Uriarte et al.* [30]. In addition, four recursive feature elimination methods, wrapped with Random Forest, were combined as follows: **RFE-RF** without any pre-filtering step (i.e. other FS methods), PCA combined with **RFE-RF**, univariate correlation filtering (**X2**) combined with **RFE-RF,** and finally, all methods were used sequentially: **X2-PCA-RFE-RF** or **X2-CM-RFE-RF**.

Fig 5 shows the performance evaluation (for the expression datasets 1, 3–7) of each complete FS combination (**X2-PCA-RFE-RF** and **X2-CM-RFE-RF**) and the random forest classification without FS step. We use the approach previously reported by *Pochet et al.* [31], where 20-fold randomized test data were used to summarize the accuracy in the prediction (see detailed description in S2 File). Also, we kept a 10-fold internal cross-validation step in all implementations of recursive feature selection trials. The results shown that when any of the full FS approaches are applied the average accuracy is higher compare with the results when not FS is used (red box plots). Only, in **Dataset 3** the workflow using PCA is less efficient than the random forest without FS step which can be related with the low number of samples analyzed (44). Importantly, even when RF perform very well it retains all the original features on each making difficult to decided which features are more relevant for the classification (S1

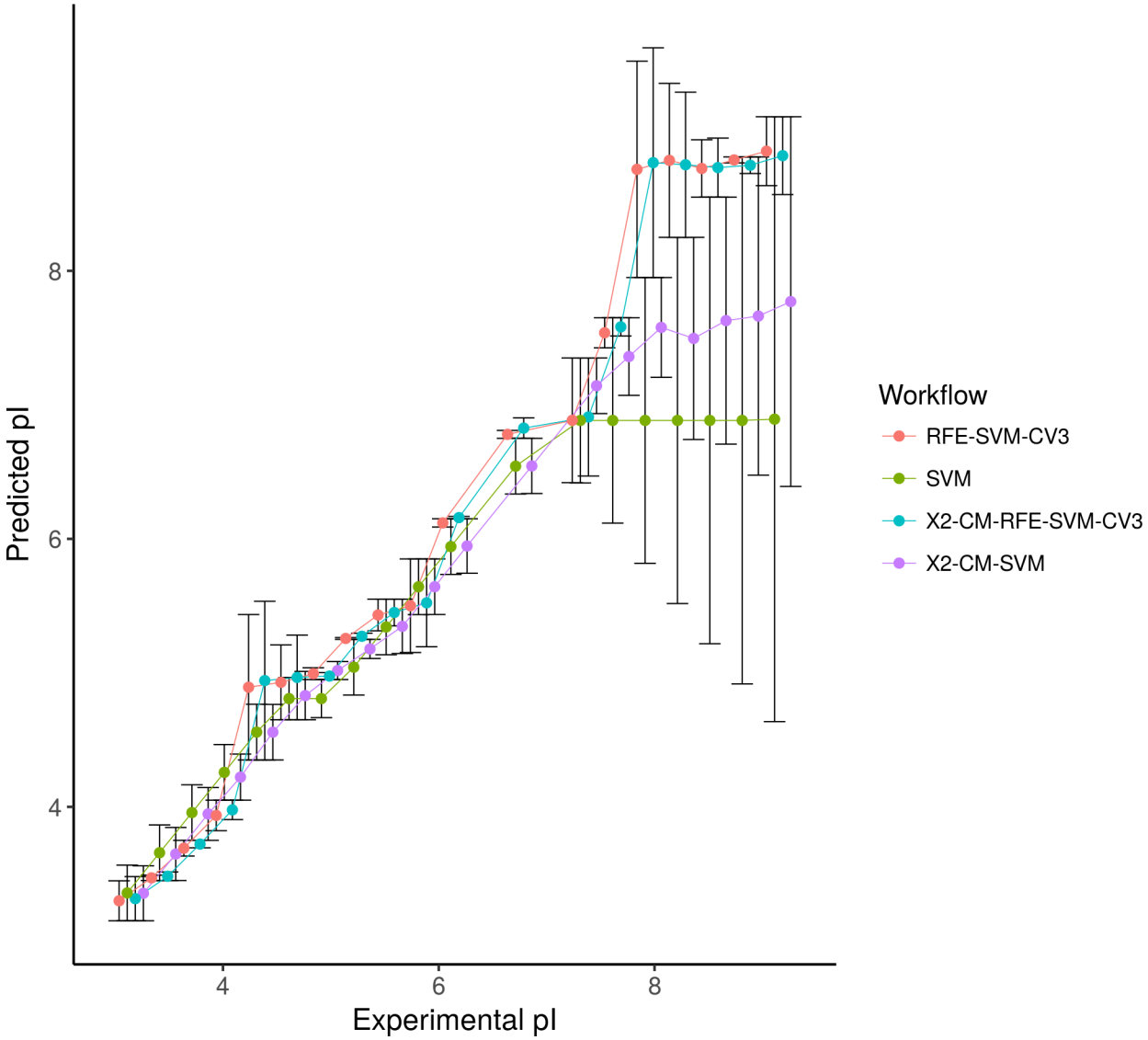

Fig 4. Error plot of predicted isoelectric point vs the experimental isoelectric point (Dataset 2): (SVM) applying FS or cross-correlation step; (X2-CM-SVM) adding correlation filters as the only steps for feature selection; (RFE-SVM-CV3) recursive feature elimination, three interactions of cross-validation combined with SVM; (X2-CM-RFE-SVM-CV3) considering the full FS workflow.

https://doi.org/10.1371/journal.pone.0189875.g004

File, Table 2). Both FS workflows reduce the number of variables in all cases in more than 90% (S1 File, Table 2), with average accuracy always above 70% (Fig 5). Because both workflow shows similar performance and some users may want to select PCA (less variables) or CM (original features), the R-package allows to define which multivariate option use during the FS.

Table 2 summarizes the benchmark metrics (accuracy, standard deviation, number of final features and time) for each evaluated FS workflow (in **Dataset 1**). While all methods kept the accuracy in the range 83–88%, when all methods were combined (proposed workflow) a lower standard deviation was obtained. Using a Random Forest model without FS, the classification process was faster than in the case of any other combination, keeping all the relevant features (1,969 of them). Including PCA and Recursive Feature Elimination (**PCA-RFE-RF**), we observed a strong feature reduction (7–10 components) and a better standard deviation (5.4).

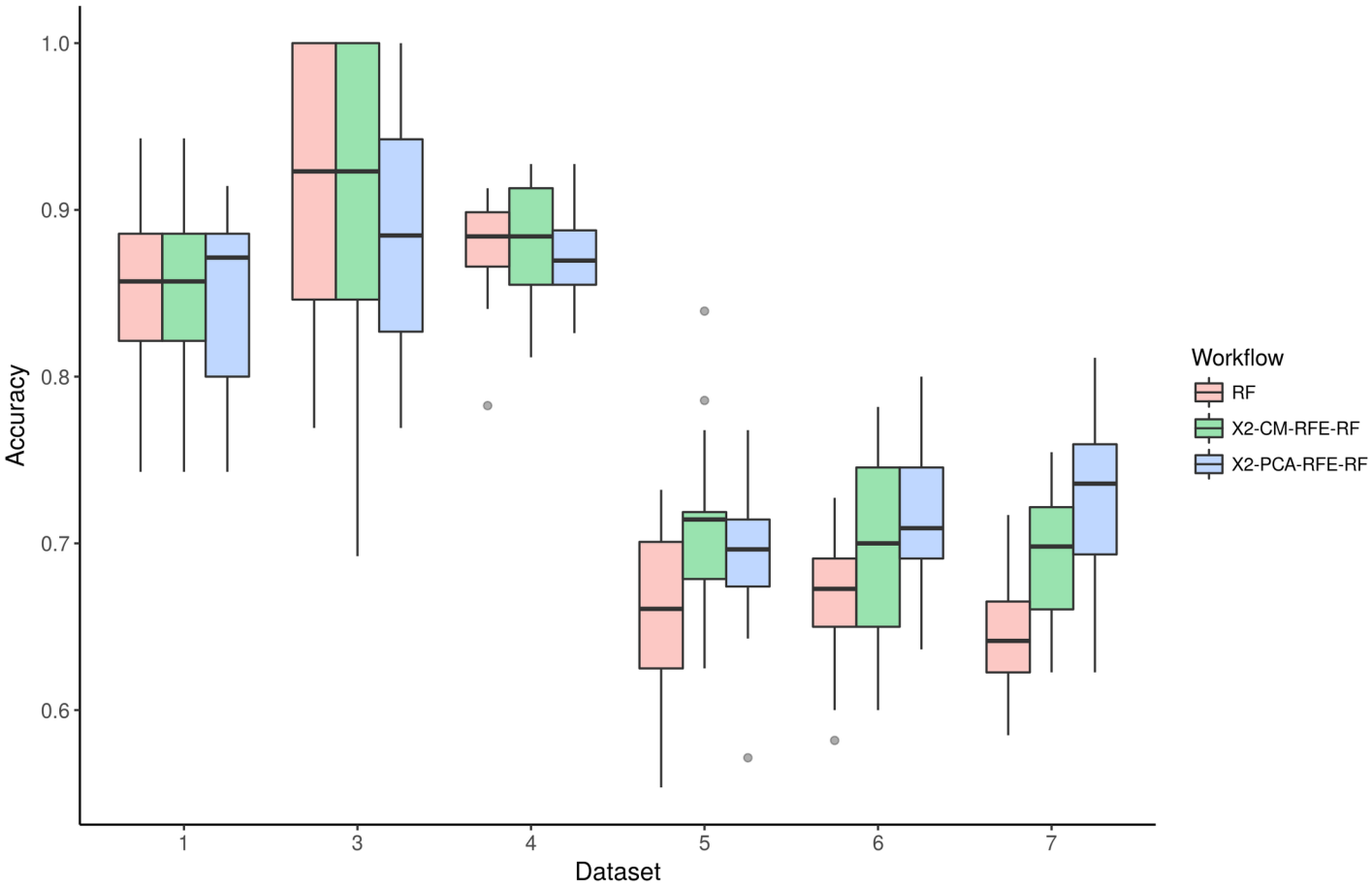

Fig 5. Accuracy vs. feature selection combination for expression datasets (1, 3, 4, 5, 6 and 7). (**RF**) Random Forest without previous feature selection step; (**X2-CM-RFE-RF**), random forest classification after the feature selection step using univariate correlation filter with matrix correlation and recursive feature elimination; (**X2-PCA-RFE-RF**), random forest classification after the feature selection step using univariate correlation filter with principal component analysis and recursive feature elimination. All methods include an internal cross-validation 10-fold step. All accuracy metrics were estimated following the approach previously reported by *Pochet et al.* [31], where 20-fold randomized test data were used to summarize the accuracy of the FS combination.

https://doi.org/10.1371/journal.pone.0189875.g005

Selecting a univariate correlation filter (**X2-RFE-RF**), a lowest standard deviation was obtained (3.6).

Fig 6 visualizes the results of the Random Forest classification algorithm without (Fig 6A, Fig 6C and Fig 6E) and with (Fig 6B, Fig 6D and Fig 6F) a FS step; for Datasets 1, 3, and 4,

Table 2. Benchmarking of the random forest model (classification) for Dataset 1, when different FS methods are applied: (RF) random forest only, (RFE) wrapper recursive feature elimination with 10-times internal cross-validation, (PCA) principal component analysis, (X2) univariate correlation filtering or (CM) correlation matrix filter. Each method is applied 20 times with randomized and class-balanced training datasets. The accuracy values provided correspond to the average value.

	Accuracy (%)	SD	Time (min)	# features
RF	83.46	8.1	1.46	1969
RFE-RF	84.61	6.3	15.83	30
PCA-RFE-RF	83.43	5.4	3.12	10
X2-RFE-RF	87.04	3.6	4.92	25
X2-PCA-RFE-RF	88.21	4.5	3.51	8
X2-CM-RFE-RF	85.01	5.7	6.35	8

https://doi.org/10.1371/journal.pone.0189875.t002

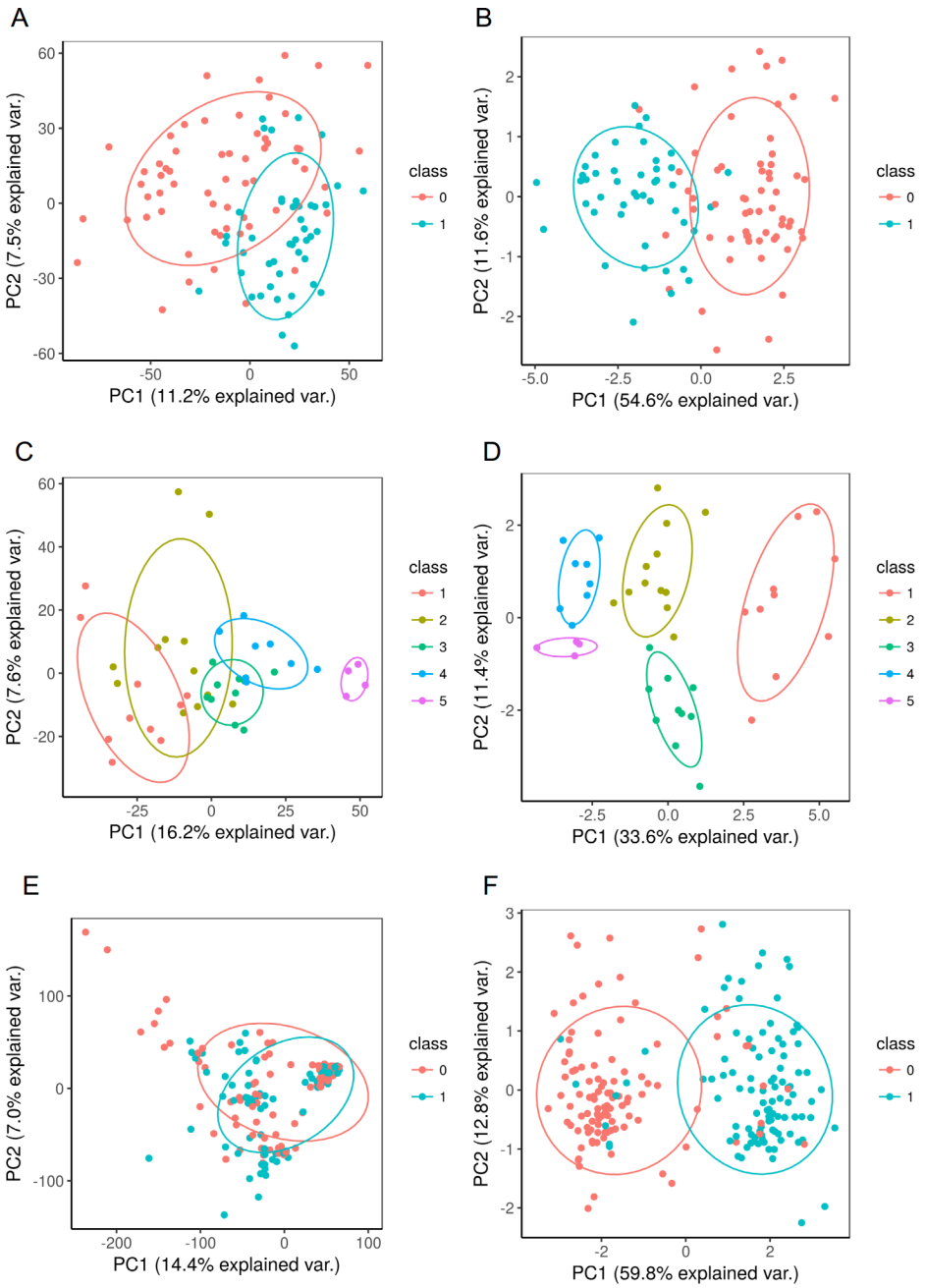

Fig 6. Visualization of the classification process using the first two principal components (PC1 and PC2) from the original data before (A, C, E) and after (B, D, F), to apply the following FS workflow: Univariate correlation (X2) with correlation matrix filter (CM) follow by Recursive Feature Elimination (RFE) wrapped with random forest (RF). The figure shows the classes distribution for **Dataset 1** (A, B), **Dataset 3** (C, D) and **Dataset 4** (E, F).

https://doi.org/10.1371/journal.pone.0189875.g006

respectively. The results show that the remaining features obtained allow to 'discriminate' between the different samples classes or groups (see detailed description in S2 File). It can be concluded that for those classification problems where the original features are needed, the PCA step could be removed without sacrificing general performance (accuracy, standard deviation, or CPU time). In contrast, univariate correlation filtering FS steps had a key impact on

Table 3. Performance comparison between the proposed approach (X2-PCA-RFE-RF) and the method reported by Li *et al.* [22]. The computer used in the original manuscript was an Intel(R) Core(TM) i5-4690 @ 3.5 GHz CPU, with 16 GB of RAM. In this study, we used an Intel(R) Core(TM) i5-4200 @ 2.5 GHz CPU, with 16 GB of RAM.

Dataset	Method	Accuracy	Variables	Runtime (min)
GSE6919/GPL8300	Current Workflow	0.77	35	8.50
	Li *et al.*	0.72	92	74.30
GSE6919/GPL92	Current Workflow	0.80	5	9.11
	Li *et al.*	0.73	174	71.50
GSE6919/GPL93	Current Workflow	0.81	6	12.00
	Li *et al.*	0.71	121	68.60

https://doi.org/10.1371/journal.pone.0189875.t003

the final results of the Random Forest model by increasing the performance in all the studied combinations. As we pointed out earlier, PCA 'obfuscates' the primary information, and thus, can potentially result in problems. When it is desirable to keep the "initial nature" of the variables, filtering methods (e.g. univariate correlation filter) exhibit a good performance (**Tables 1 and 2**) with a considerable lower number of features.

Summary of the benchmarking process

We have demonstrated the impact of the FS workflow in the classification and/or regression results as well as in the performance of the ML algorithm (CPU time and memory). Finally, we applied the same FS workflow to gene expression data from normal and prostate tumor tissues (**Datasets 5, 6 and 7**), and compared them with the results obtained by *Li et al.* [22], who used a similar approach on the same datasets (see Table 9 in [22]). Even though we observe a slight improvement in the classification accuracy in these three datasets (**Table 3**), the most notable differences were found in the number of features obtained by the final models and in the total runtime, using a similar computational platform. Thus, the results from the comparison reinforce our previous observations and validate the effectiveness of the FS workflow proposed in this manuscript. Another comparison was performed using the recently published tool based on maximum relevance–maximum distance (MRMD, http://lab.malab.cn/soft/MRMD/index_en.html) by *Zou et. al.* [32] (**Table 3**, **S1 File**). In general, we observed that both methods were comparable regarding the accuracy of the classification. However, some notable differences arose considering the number of the optimal (final) variables and the runtime. The proposed FS workflow performed better than MRMD for the analyzed datasets, by selecting in all cases less than 10% of the variables, at more than 80% reduction of the compute time.

Conclusions

FS selection algorithms are playing a major role to select correct variables for different classification and regression problems. Nevertheless, choosing the appropriate algorithm (or combination of algorithms) is not a trivial task. Different studies have highlighted methods to perform FS, but unfortunately, a thorough comparison including proper benchmarking is still lacking. Another major challenge remains: how to efficiently combine different FS methods to improve the final results. The developed FS workflow shown in this manuscript combines major strengths of univariate filtering methods, with CM and PCA strategies, as well as recursive feature elimination in two well-known learning problems: classification and regression. When univariate filtering was used in both types of problems the number of features was reduced by 80% without compromising the accuracy of the final model, and decreasing the CPU time of the learning model steps. The introduction of a wrapper method (recursive feature elimination) in combination with the learning model improved the accuracy in both

cases. If the wrapper method is applied without a previous filtering step, the CPU-time becomes too high. Finally, we demonstrated that the use of an intermediate FS step to remove redundancy between variables and features can significantly increase the accuracy of the learning model. This can be achieved by transforming the original variables into new components (retaining most of the variability in the original values) using PCA or by removing redundant highly correlated variables.

Large efforts have taken place in recent years to adopt individual FS methods. However, in our opinion, a multiple FS step workflow offers more promising results. Future developments should focus on other fields where the number of samples is growing considerably (e.g. clinical genomics, text and literature mining), and on the combination of heterogeneous datasets from different sources.

Supporting information

S1 File. Supplementary Information 1.
(DOCX)

S2 File. Supplementary Information 2.
(PDF)

Author Contributions

Conceptualization: Yasset Perez-Riverol.

Formal analysis: Yasset Perez-Riverol, Enrique Audain.

Software: Enrique Audain.

Supervision: Yasset Perez-Riverol.

Visualization: Enrique Audain.

Writing – original draft: Yasset Perez-Riverol, Enrique Audain.

Writing – review & editing: Max Kuhn, Juan Antonio Vizcaíno, Marc-Phillip Hitz.

References

1. Lynch C. Big data: How do your data grow? Nature. 2008; 455(7209):28–9. https://doi.org/10.1038/455028a PMID: 18769419.
2. Perez-Riverol Y, Bai M, da Veiga Leprevost F, Squizzato S, Park YM, Haug K, et al. Discovering and linking public omics data sets using the Omics Discovery Index. Nature biotechnology. 2017; 35 (5):406–9. https://doi.org/10.1038/nbt.3790 PMID: 28486464.
3. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007; 23(19):2507–17. https://doi.org/10.1093/bioinformatics/btm344 PMID: 17720704.
4. Barbu A, She Y, Ding L, Gramajo G. Feature Selection with Annealing for Computer Vision and Big Data Learning. IEEE Trans Pattern Anal Mach Intell. 2017; 39(2):272–86. https://doi.org/10.1109/TPAMI.2016.2544315 PMID: 27019473.
5. Perez-Riverol Y, Audain E, Millan A, Ramos Y, Sanchez A, Vizcaino JA, et al. Isoelectric point optimization using peptide descriptors and support vector machines. Journal of proteomics. 2012; 75(7):2269–74. https://doi.org/10.1016/j.jprot.2012.01.029 PMID: 22326964.
6. Wang R, Perez-Riverol Y, Hermjakob H, Vizcaino JA. Open source libraries and frameworks for biological data visualisation: A guide for developers. Proteomics. 2014. https://doi.org/10.1002/pmic.201400377 PMID: 25475079.
7. Bellman R. Dynamic programming and Lagrange multipliers. Proceedings of the National Academy of Sciences. 1956; 42(10):767–9.

8. Michalak K, Kwaśnicka H. Correlation-based feature selection strategy in classification problems. International Journal of Applied Mathematics and Computer Science. 2006; 16:503–11.
9. Wang Y, Tetko IV, Hall MA, Frank E, Facius A, Mayer KF, et al. Gene selection from microarray data for cancer classification—a machine learning approach. Computational biology and chemistry. 2005; 29 (1):37–46. https://doi.org/10.1016/j.compbiolchem.2004.11.001 PMID: 15680584.
10. Wang Y, Makedon F, Pearlman J. Tumor classification based on DNA copy number aberrations determined using SNP arrays. Oncol Rep. 2006; 15 Spec no.:1057–9. PMID: 16525700.
11. Jolliffe I. Principal component analysis: Wiley Online Library; 2002.
12. Kohavi R, John GH. Wrappers for feature subset selection. Artificial intelligence. 1997; 97(1–2):273–324.
13. Ringner M. What is principal component analysis? Nature biotechnology. 2008; 26(3):303–4. https://doi.org/10.1038/nbt0308-303 PMID: 18327243.
14. Yeung KY, Ruzzo WL. Principal component analysis for clustering gene expression data. Bioinformatics. 2001; 17(9):763–74. PMID: 11590094.
15. Yang ZR. Biological applications of support vector machines. Brief Bioinform. 2004; 5(4):328–38. PMID: 15606969.
16. Saal LH, Johansson P, Holm K, Gruvberger-Saal SK, She QB, Maurer M, et al. Poor prognosis in carcinoma is associated with a gene expression signature of aberrant PTEN tumor suppressor pathway activity. Proceedings of the National Academy of Sciences of the United States of America. 2007; 104 (18):7564–9. https://doi.org/10.1073/pnas.0702507104 PMID: 17452630; PubMed Central PMCID: PMCPMC1855070.
17. Duffy MJ. Estrogen receptors: role in breast cancer. Crit Rev Clin Lab Sci. 2006; 43(4):325–47. https://doi.org/10.1080/10408360600739218 PMID: 16769596.
18. Perez-Riverol Y, Sanchez A, Ramos Y, Schmidt A, Muller M, Betancourt L, et al. In silico analysis of accurate proteomics, complemented by selective isolation of peptides. Journal of proteomics. 2011; 74 (10):2071–82. https://doi.org/10.1016/j.jprot.2011.05.034 PMID: 21658481.
19. Audain E, Ramos Y, Hermjakob H, Flower DR, Perez-Riverol Y. Accurate estimation of isoelectric point of protein and peptide based on amino acid sequences. Bioinformatics. 2016; 32(6):821–7. https://doi.org/10.1093/bioinformatics/btv674 PMID: 26568629.
20. Lawrence RT, Perez EM, Hernandez D, Miller CP, Haas KM, Irie HY, et al. The proteomic landscape of triple-negative breast cancer. Cell Rep. 2015; 11(4):630–44. https://doi.org/10.1016/j.celrep.2015.03.050 PMID: 25892236; PubMed Central PMCID: PMCPMC4425736.
21. Wang JJ, Rau C, Avetisyan R, Ren S, Romay MC, Stolin G, et al. Genetic Dissection of Cardiac Remodeling in an Isoproterenol-Induced Heart Failure Mouse Model. PLoS genetics. 2016; 12(7):e1006038. https://doi.org/10.1371/journal.pgen.1006038 PMID: 27385019; PubMed Central PMCID: PMCPMC4934852.
22. Li S, Oh S. Improving feature selection performance using pairwise pre-evaluation. BMC bioinformatics. 2016; 17:312. https://doi.org/10.1186/s12859-016-1178-3 PMID: 27544506; PubMed Central PMCID: PMCPMC4992252.
23. Kuhn M. Caret package. Journal of Statistical Software. 2008; 28(5):1–26.
24. Liaw A, Wiener M. Classification and regression by randomForest. R news. 2002; 2(3):18–22.
25. Zeileis A, Hornik K, Smola A, Karatzoglou A. kernlab-an S4 package for kernel methods in R. Journal of statistical software. 2004; 11(9):1–20.
26. Romanski P, Kotthoff L, Kotthoff ML. Package 'FSelector'. URL http://cran/r-project.org/web/packages/FSelector/index.html; 2013.
27. Audain E, Sanchez A, Vizcaino JA, Perez-Riverol Y. A survey of molecular descriptors used in mass spectrometry based proteomics. Current topics in medicinal chemistry. 2014; 14(3):388–97. PMID: 24304317.
28. Chambers SE, Hoskins PR, Haddad NG, Johnstone FD, McDicken WN, Muir BB. A comparison of fetal abdominal circumference measurements and Doppler ultrasound in the prediction of small-for-dates babies and fetal compromise. Br J Obstet Gynaecol. 1989; 96(7):803–8. PMID: 2669932.
29. Varma S, Simon R. Bias in error estimation when using cross-validation for model selection. BMC bioinformatics. 2006; 7:91. https://doi.org/10.1186/1471-2105-7-91 PMID: 16504092; PubMed Central PMCID: PMCPMC1397873.
30. Diaz-Uriarte R, Alvarez de Andres S. Gene selection and classification of microarray data using random forest. BMC bioinformatics. 2006; 7:3. https://doi.org/10.1186/1471-2105-7-3 PMID: 16398926; PubMed Central PMCID: PMCPMC1363357.

31. Pochet N, De Smet F, Suykens JA, De Moor BL. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. Bioinformatics. 2004; 20 (17):3185–95. https://doi.org/10.1093/bioinformatics/bth383 PMID: 15231531.

32. Zou Q, Zeng J, Cao L, Ji R. A novel features ranking metric with application to scalable visual and bioinformatics data classification. Neurocomputing. 2016; 173:346–54.