ROOFS: RObust biOmarker Feature Selection

Anastasiia Bakhmach, Paul Dufossé, Andrea Vaglio, Florence Monville, Laurent Greillier, Fabrice Barlési, Sébastien Benzekry

ROOFS: ROBUST BIOMARKER FEATURE SELECTION

Anastasiia Bakhmach[1], Paul Dufossé[1], Andrea Vaglio[1], Florence Monville[2], Laurent Greillier[1,3], Fabrice Barlési[4], and Sébastien Benzekry[1]

[1]Inria – Inserm team COMPO, COMPutational pharmacology and clinical Oncology, Centre Inria Sophia Antipolis - Méditerranée, Centre de Recherches en Cancérologie de Marseille, Inserm U1068, CNRS UMR7258, Institut Paoli-Calmettes, Pharmacy faculty, Aix-Marseille University
[2]Veracyte SAS, Marseille, France
[3]Assistance Publique-Hôpitaux de Marseille (APHM), Marseille, France
[4]Gustave Roussy, Villejuif, France

January 9, 2026

ABSTRACT

Feature selection (FS) is essential for biomarker discovery and in the analysis of biomedical datasets. However, challenges such as high-dimensional feature space, low sample size, multicollinearity, and missing values make FS non-trivial. Moreover, FS performances vary across datasets and predictive tasks. We propose *roofs*, a Python package available at `https://gitlab.inria.fr/compo/roofs`, designed to help researchers in the choice of FS method adapted to their problem. *roofs* benchmarks multiple FS methods on the user's data and generates reports that summarize a comprehensive set of evaluation metrics, including downstream predictive performance estimated using optimism correction, stability, reliability of individual features, and true positive and false positive rates assessed on semi-synthetic data with a simulated outcome. We demonstrate the utility of *roofs* on data from the PIONeeR clinical trial, aimed at identifying predictors of resistance to anti-PD-(L)1 immunotherapy in lung cancer. The PIONeeR dataset contained 374 multi-source blood and tumor biomarkers from 435 patients. A reduced subset of 214 features was obtained through iterative variance inflation factor pre-filtering. Of the 34 FS methods gathered in *roofs*, we evaluated 23 in combination with 11 classifiers (253 models in total) and identified a filter based on the union of Benjamini-Hochberg false discovery rate-adjusted p-values from t-test and logistic regression as the optimal approach, outperforming other methods including the widely used LASSO. We conclude that comprehensive benchmarking with *roofs* has the potential to improve the robustness and reproducibility of FS discoveries and increase the translational value of clinical models.

1 Introduction

Feature selection (FS) refers to a class of methods aimed at reducing the feature space by selecting a subset of size $k < p$. In supervised learning, FS is used either for discovery (i.e., to identify the most informative variables), or for prediction, to develop models with reduced variance. A basic requirement for FS is that the reduced model should perform equivalently to or better than the full model, capturing the complexity of the original data. In the context of biomarker discovery and clinical predictive models, the selected signature, defined as a minimal set of variables with high predictive capacity, should meet additional requirements: (i) contain biologically or clinically interpretable features; (ii) be as minimal as possible; and (iii) demonstrate *stability*, i.e. the selected features should not be sensitive to small changes in the data [1]. FS stability is particularly critical for ensuring reproducibility of biomarker signatures. It is typically assessed by measuring the similarity between feature subsets obtained across resamples of the original data, e.g. via bootstrapping or subsampling. A method that is not able to reproduce similar feature subsets across resamples is unlikely to capture a reliable signature in the original dataset and may result in models that do not generalize well to external populations. Consequently, risk prediction models or biomarker panels built on unstable signatures are unlikely

to be clinically useful. Importance of stability evaluation has been frequently acknowledged in the literature [1, 2, 3] but is yet often neglected in practice.

The complexity of biomedical data has several important implications for FS. First, the number of samples (here, number of patients, denoted n) is often smaller than the number of features (denoted p), leading to ill-posedness. Second, the often-encountered high degree of multicollinearity makes FS prone to instability because correlated features can be used interchangeably by a model. Third, high rates of missing values are common, which further complicates the discovery, increasing the rate of false positives and false negatives [4].

A wide range of FS methods has been proposed in the literature (see [5] for an extensive review). Briefly, based on the underlying algorithmic approach, FS methods can be classified into 4 groups: filter, embedded, wrapper, and ensemble methods. Filters evaluate features according to a specific criteria and output either a complete ranking or a subset selected using a predefined threshold (e.g., univariable or multivariable statistical tests). Embedded methods perform FS simultaneously with model learning, as in regularization-based approaches or tree-based models with constrained depth. Wrapper methods iteratively search for feature subsets that maximize the predictive performance of a given model. Ensemble methods aim to improve the robustness of FS by aggregating outputs from different selection algorithms or from repeated runs of a single method on resampled data. Filter and embedded methods are typically more straightforward, computationally efficient, and interpretable compared to wrapper and ensemble techniques. At the same time, previous benchmarks [6, 7] have shown that, in terms of predictive performances, the optimal method is contingent upon the specific dataset, and no FS method consistently outperformed others across diverse predictive tasks. Furthermore, FS performance is sensitive to hyperparameter choice, such as the selected subset size, which could alter the relative ranking of methods within the same dataset [8].

Given the numerous challenges in FS for biomedical data highlighted above, researchers may be hesitant about which FS approach to use when analyzing a new dataset. To address this gap, we propose a Python package called *roofs* available at https://gitlab.inria.fr/compo/*roofs*. *roofs* conducts comprehensive benchmarking of multiple pre-implemented FS algorithms on the user's data based on three criteria: 1) method stability and robustness of individual features, 2) predictive performances, and 3) true positive and false positive rates (from semi-synthetic data with a simulated outcome). For point 2), we relied on the optimism-correction (OC) framework emphasized as a guideline for development of clinical predictive models by Collins et al. [9]. *roofs* provides automated reporting of the evaluation metrics listed above – a key feature of the package aimed to guide users in selecting the FS approach adapted to their dataset.

We demonstrate the practical value of *roofs* by applying it to a complex dataset from the PIONeeR clinical study (NCT03833440, Precision Immuno-Oncology for Advanced Non-small Cell Lung Cancer Patients With PD-1 ICI Resistance) to predict the resistance to immunotherapy. This dataset is an interesting and challenging use case for FS because: 1) the data is rich and heterogeneous combining clinical features, routine blood tests and detailed immunoprofiling of blood and tumor samples measured by different clinical and research partners; 2) it contains clusters of highly correlated features, e.g. numerous variables describing related immune cell populations; and 3) p (374) is roughly equivalent to n (435). This is different from the three types of datasets popular in the literature: a) high-dimensional omics ($p \gg n$); b) data with a sufficiently high event-per-variable ratio, e.g. the number of patients in the smallest class is at least ten times the number of predictors $n_{min} \geq 10p$ (for classification problems); and c) big data (e.g., from electronic health records), typically modeled using deep artificial neural networks.

A comprehensive survey of FS benchmarks across multiple domains published before 2020 is provided by Hopf and Reifenrath [10]. Table 1 summarizes more recent benchmarks on high-dimensional biomedical data. Most of these studies focused on a single data modality (genomic data). Only two studies included FS methods representing all algorithmic families, and stability was evaluated in only three of the eight benchmarks. The strengths of our study on PIONeeR are the following: 1) it addresses heterogeneous, mixed-type data; 2) it evaluates a broader and more diverse range of FS algorithms; 3) it reports all standard performance metrics, with an emphasis on stability; and 4) it extends FS evaluation to assess the balance between true and false discoveries.

To summarize our contribution:

1) In the Methods section, we introduce a Python framework for comprehensive evaluation of FS methods on user-provided data to help identify the most suitable FS approach for a given dataset and predictive task.

2) In the Results section, we demonstrate the applicability of this framework using a clinical dataset from lung cancer patients comprising diverse biomarkers, including demographic, clinical, and immunoprofiling data from both blood and tumor samples, to improve the prediction of resistance to immunotherapy.

Reference	Data type	#D	#M	FS Methods				Metrics		
				F	E	W	Ens	Predictive performance	Stability	Runtime
Cattelani et al. [11]	Genomics	8	7		✓	✓		✓	✓	
Demircioğlu [7]	Radiomics	50	9	✓	✓	✓		✓		✓
Labory et al. [12]	Omics	3	5	✓		✓		✓		
Budhraja et al. [13]	Omics	4	11	✓	✓	✓	✓	✓	✓	
Li et al. [14]	Cancer multi-omics	15	8	✓	✓	✓		✓		✓
Bommert et al. [15]	Gene expression survival data	11	14	✓				✓	✓	✓
Bhadra et al. [16]	Cancer multi-omics	5	5	✓	✓	✓		✓		
Chen et al. [17]	Not specified	4	12	✓	✓	✓	✓	✓		
Our work	**Clinical + biological**	**1**	**23**	✓	✓	✓	✓	✓	✓	✓

#D = number of datasets, #M = number of FS methods, F = filters, E = embedded methods, W = wrappers, Ens = ensemble methods

Table 1: Recent feature selection benchmarks on high-dimensional biomedical data

2 Methods

2.1 Data

2.1.1 PIONeeR dataset

The development of *roofs* was motivated by the following real-world problem: identifying robust biomarkers of resistance to immunotherapy in lung cancer. To this end, we relied on the PIONeeR clinical trial dataset, derived from a prospective, multicenter cohort of patients (17 centers in France) with advanced or recurrent non-small cell lung cancer (NSCLC). Patients received either frontline combination therapy with platinum-based chemotherapy plus anti-PD-(L)1 immune checkpoint inhibitors (ICIs) or second-/third-line monotherapy with anti-PD-(L)1 ICIs following progression on prior platinum-based chemotherapy. The dataset integrated multi-modal data from $n = 435$ patients, comprising a total of $p = 374$ variables. It included clinical and demographic variables (e.g., sex, age, $p = 10$), routine blood tests (e.g., blood counts, biochemistry, $p = 49$), tumor multiplex immunohistochemistry markers (e.g., immune cells, $p = 159$), and circulating immune and vasculophenotyping markers. The latter included both flow cytometry data ($p = 141$) and soluble ($p = 15$) markers. The dataset contained 30.8% of missing values. The outcome to predict was primary resistance to immunotherapy, defined as disease progression within 6 months of treatment [18].

2.1.2 Reduced PIONeeR dataset

To address multicollinearity, a reduced dataset with 214 features was derived from the full data by applying iterative variance inflation factor (VIF) pre-filtering [19], sequentially removing the feature with the highest VIF until all remaining features had VIF ≤ 5. At each iteration, VIF of the j_{th} feature was computed as:

$$\text{VIF}_j = \frac{1}{1 - R_j^2}$$

where R_j^2 stands for the coefficient of determination from regressing the j_{th} feature onto all remaining predictors.

2.1.3 Simulated outcome

To evaluate each method's ability to identify the "true features" (i.e., the ones hypothesized as having generated the outcome), a part of FS benchmarking implemented in *roofs* is conducted on a semi-synthetic dataset with a simulated outcome denoted $\tilde{y}$. With the true model being unknown, the real outcome y cannot be used directly for this purpose, but the original $n \times p$ features matrix X can still be used. The simulated outcome was derived by applying a logistic function to a linear combination of S pre-selected features from the real dataset $\mathbf{X}$ with added Gaussian noise:

$$\text{score}^i = \frac{1}{1 + \exp\left(-\left(\sum_{j \in \mathcal{S}} \beta_j X_j^i + \varepsilon\right)\right)}, \quad \varepsilon \sim \mathcal{N}(0, 1)$$

$$\tilde{y}^i = \begin{cases} 1 & \text{if score}^i \geq 0.5 \\ 0 & \text{otherwise} , \end{cases}$$

where $\mathcal{S}$ is the index set of the specified true predictors and β_j is the coefficient of j^{th} feature estimated from an unpenalized logistic regression fitted using this set of predictors.

For the benchmark on PIONeeR data, the true predictors (19 features in total) were selected from the full dataset using a 0.01 threshold on adjusted p-values from a multivariable Cox model for progression-free survival. This approach was chosen to mimic a realistic data structure.

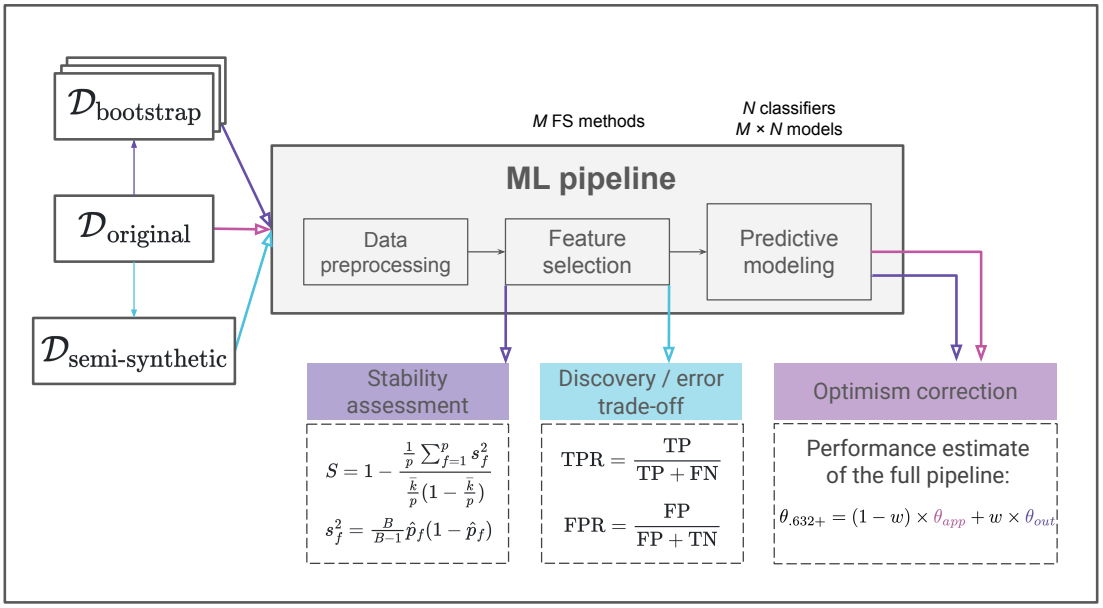


Figure 1: Overview of the *roofs* pipeline for comprehensive FS benchmarking.

2.2 Suggested framework

The evaluation of a given FS method by *roofs* is based on a bootstrap experiment (Figure 1). Let $\mathcal{D}_{\text{original}} = \{(x^i, y^i)\}_{i=1}^n$ denote the original dataset provided by the user. From this, B bootstrap resamples are derived, each denoted $\mathcal{D}_b$, $b \in [1, B]$. In the benchmark on PIONeeR, the number of resamples was set to 100. An identical pipeline, including data preprocessing (imputation using the median for continuous variables or mode for categorical variables and z-score normalization), FS and subsequent predictive modeling, is applied to both the full data and each bootstrap, and repeated across all user-specified combinations of FS methods and classifiers.

2.2.1 FS methods

roofs gathers 34 FS methods from all algorithmic families – filters, embedded methods, wrappers, and ensembles – including both classical statistical techniques and recent ML-based approaches. Most implementations were adapted from existing Python libraries, including scikit-learn [20], scikit-feature [5], Stabl [21], and others.

Of the 34 FS methods implemented in *roofs*, 23 representative approaches were included in the present benchmark. Among embedded methods, we evaluated LASSO [22] and its variants (modified Bolasso [23] with a frequency threshold of 0.5 followed by a second LASSO fit, adaptive LASSO [24], exclusive LASSO [25], and HSIC LASSO [26]), as well as ensemble bootstrap-based methods designed to enhance stability and control false discoveries (stability selection [27], Stabl LASSO [21], RENT [28]). Both univariate and multivariate filters were used, including information theory (CIFE [29], CMIM [30], DISR [31], JMI [32]) and similarity-based approaches (ReliefF [33], Fisher score [5]),

correlation-based clustering (hierarchical clustering), and statistical methods (Gini index, t-score, minimum redundancy maximum relevance (mRMR) [5], and adjusted p-values from Benjamini-Hochberg method [34]). Wrappers included two recursive elimination methods (LR-RFE, based on logistic regression coefficients, and RF-RFE, based on random forest importances), a forward selection method (forward RF), and the SHAPley-based Shapicant [35]. Finally, random feature selection was included as a control. A complete description of all benchmarked methods is provided in Supplementary Table S1.

2.2.2 Classifiers

roofs evaluates the predictive performance of signatures resulting from the FS step by training downstream classifiers on the corresponding post-selection datasets. *roofs* allows using any scikit-learn-compatible classifier. Since predictive performances can depend on the interactions between the FS method and the classifier, it is recommended to test a wide range of modelling approaches, such as linear models, tree-based methods, and ensembles. The classifiers used in the present benchmark study are listed in Supplementary Table S2.

2.2.3 Optimism correction

roofs uses optimism correction (OC) for evaluation of the predictive performances, which allows not to lose part of the data due to train/test splitting [9]. Three OC methods were implemented: Harrell [36], .632 [37], and .632+ [38]. The following set of metrics are calculated: AUC, accuracy, sensitivity, specificity, positive predictive value, negative predictive value.

In Harrell's classical method, optimism is defined as the average difference between model performance on bootstrap samples (in training) and on the original dataset (in testing):

$$O = \frac{1}{B} \sum_{b=1}^{B} \left(\theta_{\text{boot}}^{(b)} - \theta_{\text{orig}}^{(b)} \right),$$

where $\theta_{\text{boot}}^{(b)}$ is the training performance of the b-th bootstrap model, $\theta_{\text{orig}}^{(b)}$ is the test performance of the same model on the original dataset, and B is the number of bootstrap replicates.

The final optimism-corrected performance is obtained by subtracting the average optimism from the apparent performance (i.e., the training performance on the full dataset):

$$\theta_{\text{Harrell}} = \theta_{\text{app}} - O,$$

where θ_{app} denotes the apparent performance.

The .632 method accounts for the overlap between the bootstrap samples and the original data and computes θ_{orig} only on out-of-bag (OOB) samples (not included into the bootstrap, therefore not seen during training). The performance estimate is computed as

$$\theta_{.632} = 0.368 \cdot \theta_{\text{app}} + 0.632 \cdot \theta_{\text{out}},$$

where $\theta_{\text{out}} = \frac{1}{B} \sum_{b=1}^{B} \theta_{\text{OOB}}$. The weight 0.632 corresponds to the approximate proportion of samples included in a bootstrap [39].

The .632+ method improves the .632 rule by modifying the weights in order to take into account the amount of overfitting:

$$\theta_{.632+} = (1 - w) \cdot \theta_{\text{app}} + w \cdot \theta_{\text{out}},$$

where

$$w = \frac{0.632}{1 - 0.368 \cdot R}.$$

R is the overfitting ratio computed as

$$R = \frac{\theta_{\text{app}} - \theta_{\text{out}}}{\theta_{\text{app}} - \gamma},$$

where γ corresponds to the predictive performance on the original dataset when the outcome is randomly permuted. When there is no overfitting (i.e., the apparent score is the same as the out-of-bag score), R equals to 0, w equals to 0.632, and the .632+ rule replicates the .632 rule. In the presence of overfitting (apparent score is greater than the out-of-bag score), $R \in (0, 1]$ and $w \in (0.632, 1]$. In a setting of complete overfitting, w equals to 1 and $\theta_{.632+} = \theta_{\text{out}}$. The results reported in this paper were obtained using .632+ method as it has been demonstrated to have the lowest bias in multiple settings [40].

2.2.4 FS stability

The FS stability in *roofs* is calculated from the bootstrap-derived feature sets using Nogueira's frequency-based measure (see [41] for details and a review of alternative stability metrics):

$$S = 1 - \frac{\frac{1}{p}\sum_{f=1}^{p} s_f^2}{\frac{\bar{k}}{p}(1 - \frac{\bar{k}}{p})}$$

where $s_f^2 = \frac{B}{B-1}\hat{p}_f(1 - \hat{p}_f)$ is the unbiased sample variance of the frequency of selection $\hat{p}_f$ of f^{th} feature over B bootstraps and $\bar{k}$ is an average size of the selected subsets.

Additionally, the package reports simple measures of individual feature robustness. For each feature, it computes the selection frequency obtained by every FS method and identifies how many methods select that feature with high frequency (as defined by the user; we used a 50% threshold). The selection frequency serves as an indicator of the signal strength: features that are weakly associated with the outcome appear in only a small fraction of bootstrap models, whereas features with a strong signal are consistently selected in a large proportion of bootstrap models and across multiple FS methods. This enables to distinguish between features that are stable and predictive and those that may be artifacts of a specific algorithm or over- or undersampling of particular patient subclusters.

2.2.5 Trade-off between true and false discoveries

roofs additionally benchmarks FS methods on a semi-synthetic dataset with simulated outcome $\mathcal{D}_{\text{semi-synthetic}} = \{(x^i, \tilde{y}^i)\}_{i=1}^n, \quad x^i \in X, \tilde{y}^i \in \tilde{Y}$. The comparison of FS methods on a semi-synthetic data allows assessing the trade-off between true and false discoveries, since the underlying true model is known. The true positive rate (TPR) – the proportion of true predictors correctly identified – represents the discovery power of an FS algorithm. The false positive rate (FPR) – the proportion of features not included in the true model but selected by an algorithm – shows the cost of discovery in terms of the error. The procedure used to derive the synthetic outcome in our benchmark on PIONeeR dataset is described in subsection 2.1.3. *roofs* supports modeling both linear and non-linear relationships between predictors and outcome, but in the present benchmark we used a linear model for simplicity. Variability in FPR and TPR is assessed by repeating the experiment with resampled noise, with the number of replicates specified by the user (set to 36 in our benchmark).

FS methods gathered in *roofs* can be subdivided into two groups based on how they determine the number of features they select: 1) methods with fixed, user-defined subset size, and 2) methods with subset size based on internal criteria, such as cross-validated lambda in the case of LASSO or p-value thresholds in the case of p.adjust or Shapicant. In the experiment on the semi-synthetic dataset, methods from the first group were configured to select 19 features, corresponding to the number of predictors in the true model. This configuration provided these methods with an advantage compared to the second group of methods with respect to both true and false positives. Their TPR was lower bounded, as by design the selection conducted by these methods was not too stringent, and only a random or very inaccurate method would have obtained a low TPR. In turn, their FPR was upper bounded, as these methods could not select more than 19 features.

2.2.6 Automated reporting

To allow users to efficiently evaluate the results of benchmarking, *roofs* generates a report summarizing the key metrics of FS performance. The report includes: 1) a table that aggregates FS frequencies across methods and shows agreement between different methods in selecting individual features; 2) a stability – AUC plot for a quick visual overview of performance and identification of Pareto-optimal methods; 3) a summary table that combines all metrics from every model, with filtering options that help identify the optimal methods based on user's criteria of interest.

2.2.7 Choice of FS method

roofs allows users to manually choose a method that best fits their objectives based on the metrics included in the automatically generated report. By itself, the package does not select the best FS method from user's data, as the acceptable trade-off between different metrics depends on the specific prediction or discovery goals.

2.2.8 Experimental setup

All experiments on PIONeeR data were conducted on a MacBook Pro (2019) with a 2.4 GHz 8-core Intel Core i9 processor and 64 GB 2667 MHz DDR4 RAM.

3 Results

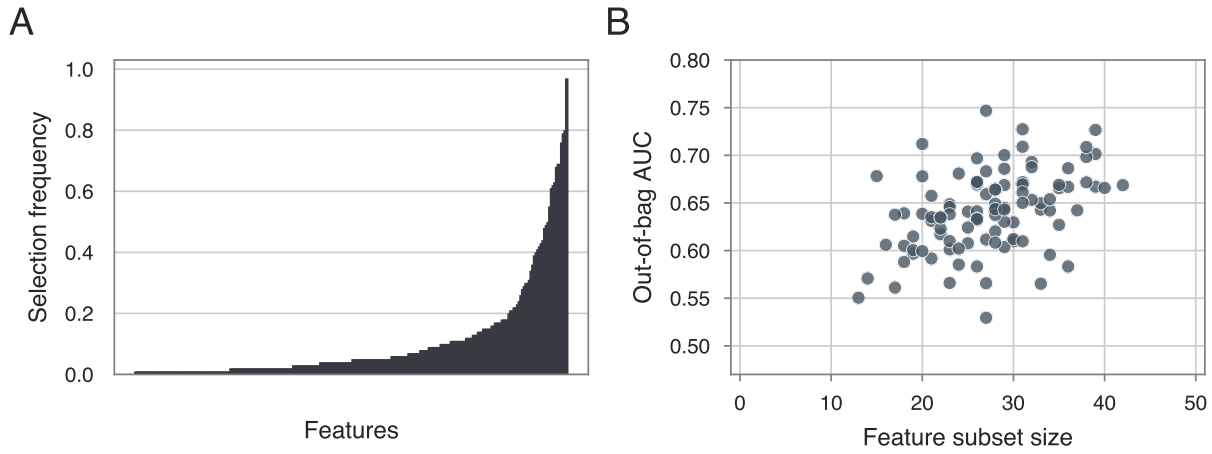


Figure 2: Instability of LASSO. **A:** Bootstrap selection frequencies for features selected by LASSO in at least 1 bootstrap sample. **B:** Variability in selected subset sizes and out-of-bag AUC.

3.1 LASSO-derived signature shows modest stability

Our motivation in the PIONeeR study was to establish a biomarker signature capable of explaining resistance to immunotherapy in advanced lung cancer. We first applied our benchmarking pipeline, as described in Methods and illustrated in Figure 1, using classical LASSO (Least Absolute Shrinkage and Selection Operator) as a baseline FS method. LASSO remains a standard FS approach in biomedical data analysis due to its simplicity of interpretation and computational efficiency, but it has been shown to be inconsistent (i.e., unable to recover a true model) in the presence of linear dependencies among features, which is common in high-dimensional problems [42]. On the PIONeeR data, LASSO showed moderate stability (S = 0.34, Figure 2A) and high variability in features selected across bootstrap datasets, with 255 out of 374 features selected in at least 1 bootstrap (Figure 2A). LASSO-selected feature sets varied substantially in size, ranging from 13 to 42 predictors, and the corresponding post-selection models, trained using classical ML classifiers, achieved out-of-bag AUC values (θ_{out} in the Methods) between 0.49 and 0.78. The apparent LASSO model selected 25 features, of which only 12 were selected with high confidence (defined here as being chosen in at least 50% of bootstrap models), and none were selected across all bootstraps. Low selection frequency of the remaining 13 features raised concerns about their potential clinical relevance and generalizability of the signature.

3.2 Iterative VIF increases FS stability

Since LASSO instability could be partially attributed to multicollinearity among predictors, we next applied iterative VIF pre-filtering to the full data (374 features) to derive a feature set with lower redundancy and evaluate its impact on FS stability. This produced a reduced dataset of 214 features (fig. 3A). The mean R^2 obtained from a regression of each feature onto all other predictors decreased from 0.91 (range: 0.41-0.99) in the full dataset to 0.63 (range: 0.22-0.79) following VIF pre-filtering.

To evaluate whether this reduction improved FS stability, we conducted two separate benchmarks of 23 FS methods on the full and reduced datasets. VIF pre-filtering led to a small increase in stability of the majority of FS methods (fig. 3B), likely driven by the fact that fewer features were used interchangeably across bootstrap models. Based on this improvement, all subsequent results are reported on the post-VIF dataset.

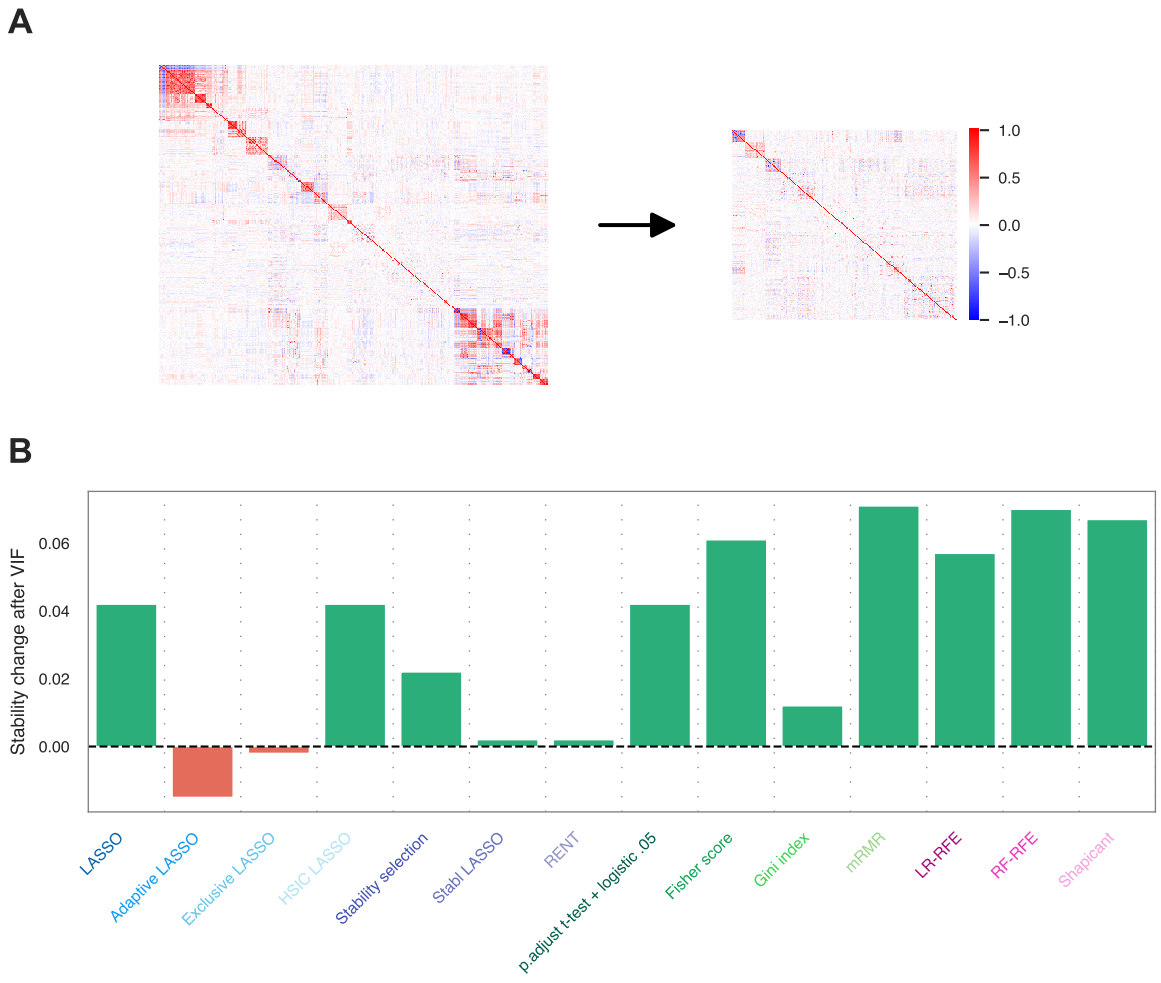


Figure 3: Improvement in FS stability following multicollinearity reduction with VIF pre-filtering compared to the original dataset. **A:** Correlation structure of the dataset before and after applying VIF pre-filtering. **B:** Absolute change in FS stability across representative FS methods.

3.3 Comprehensive benchmarking shows variability in FS stability and performance

On the reduced post-VIF dataset, filters were generally the most stable and predictive family of FS methods, although variability was observed within the group. Specifically, statistical filters achieved high optimism-corrected AUC values (ranging from 0.7 to 0.72), and several of them demonstrated good stability, for example, Fisher score (S = 0.37), t-score (S = 0.36), and p.adjust (S = 0.39) (see Table 2). In contrast, CIFE and DISR, filters based on information theory scores, achieved top stability values (S = 0.73 and 0.54, respectively) but lower predictive performance (optimism-corrected AUC = 0.67 and 0.61, respectively). All information theory filters (CIFE, CMIM, DISR, JMI) often omitted known biomarkers (e.g. lactate dehydrogenase and C-reactive protein), showing that selection can be consistent but wrong. CMIM and JMI were both unstable and not predictive, although still outperforming random FS. Embedded methods exhibited the lowest stability among other groups at comparable predictive performance. Surprisingly, embedded resampling-based ensembles designed to improve the consistency of FS (stability selection, Stabl LASSO, and RENT) did not outperform standard LASSO neither in stability nor in AUC. Their stringent selection may account for their lower predictive performance, but it does not explain the lack of improvement in stability (S = 0.3, 0.25, and 0.29,

Table 2: Results of FS benchmarking on the post-VIF PIONeeR dataset (n = 435, p = 214)

Method	AUC	Stability	FPR	TPR	Run time (s)
LASSO	0.69 ^{GB}	0.34	0.02 ± 0.01	0.59 ± 0.09	0.27 ± 0.02
Bolasso	0.69 ^{SVC}	0.34	0.01 ± 0.01	0.49 ± 0.07	17.48 ± 0.71
Adaptive LASSO	0.67 St	0.32	0.05 ± 0.01	0.59 ± 0.06	0.24 ± 0.04
Exclusive LASSO	0.66 St	0.37	0.06 ± 0.00	0.27 ± 0.03	0.59 ± 0.06
HSIC LASSO	0.69 ^{Ex}	0.25	0.06 ± 0.01	0.41 ± 0.08	0.69 ± 0.06
Stability selection	0.67 St	0.3	0.01 ± 0.01	0.46 ± 0.07	8.10 ± 0.6
Stabl LASSO	0.64 St	0.25	**0.00 ± 0.01**	0.31 ± 0.07	47.36 ± 5.2
RENT	0.69 ^{Bag}	0.29	0.01 ± 0.01	0.47 ± 0.07	46.48 ± 4.23
CIFE	0.67 ^{MLP}	**0.73**	0.08 ± 0.00	0.16 ± 0.03	3.4 ± 0.14
CMIM	0.62 ^{Bag}	0.18	0.08 ± 0.01	0.14 ± 0.07	3.62 ± 0.15
DISR	0.61 ^{Ex}	0.54	0.08 ± 0.00	0.21 ± 0.02	10.11 ± 0.23
JMI	0.6 ^{Bag}	0.18	0.08 ± 0.80	0.21 ± 0.00	3.79 ± 0.07
ReliefF	0.68 ^{MLP}	0.49	0.06 ± 0.01	0.37 ± 0.13	0.26 ± 0.03
Fisher score	0.7 ^{GB}	0.37	0.03 ± 0.01	0.69 ± 0.06	0.19 ± 0.03
Gini index	0.69 ^{Ex}	0.32	0.04 ± 0.01	0.61 ± 0.07	1.00 ± 0.04
t-score	0.7 ^{GB}	0.36	0.03 ± 0.01	0.71 ± 0.06	**0.11 ± 0.03**
mRMR	0.7 ^{LDA}	0.32	0.04 ± 0.01	0.63 ± 0.07	0.82 ± 0.03
Hierarchical clustering	0.7 ^{GB}	0.23	0.06 ± 0.00	0.38 ± 0.04	9.44 ± 0.47
p.adjust t-test + logistic .05	**0.72** ^{GB}	0.39	0.10 ± 0.03	**0.93 ± 0.06**	1.95 ± 0.09
LR-RFE	0.69 ^{LDA}	0.38	0.03 ± 0.01	0.7 ± 0.06	**0.11 ± 0.03**
RF-RFE	0.7 ^{GB}	0.38	0.02 ± 0.02	0.46 ± 0.11	52.27 ± 0.30
Forward RF	0.69 ^{LR}	0.34	0.02 ± 0.02	0.47 ± 0.12	16.29 ± 0.13
Shapicant	0.7 ^{Bag}	0.31	0.06 ± 0.01	0.71 ± 0.04	25.96 ± 0.20
Random FS (fixed size)	0.58 ^{Ex}	0	0.09 ± 0.00	0.05 ± 0.00	
Random FS		0	0.07 ± 0.03	0.08 ± 0.04	
Full data	0.71 ^{GB}	1			

Results are reported for the optimal FS method-classifier pair, representing the best performance across all tested classifiers. Values are presented as point estimates or mean ± SD (standard deviation); run times correspond to the average duration of the FS step (excluding predictive modeling) across bootstraps. Better than LASSO. GB: Gradient boosting classifier; St: stacking classifier; Ex: extra trees classifier; Bag: bagging classifier; LR: logistic regression; SVC: support vector classifier; LDA: linear discriminant analysis; MLP: multi-layer perceptron; LASSO: least absolute shrinkage and selection operator; HSIC: Hilbert-Schmidt independence criterion; RENT: repeated elastic net technique; CIFE: conditional infomax feature extraction; CMIM: conditional mutual information maximization; DISR: double input symmetrical relevance; JMI: joint mutual information; mRMR: minimum Redundancy Maximum Relevance; LR-RFE: logistic regression-based recursive feature elimination.

respectively, vs LASSO S = 0.34). The instability of these methods could be attributed to the variability behind the resampling procedures on which they rely, although this requires further investigation. The performance of the wrapper methods was intermediate, with AUC (0.69-0.70) and stability (0.32-0.38) falling between filter methods (the best) and embedded methods (the worst).

Among all methods, the p.adjust, a statistical filter that combined t-test and logistic regression adjusted p-values at a 0.05 threshold, achieved the best balance between stability (S = 0.39) and performance (AUC = 0.72) (Figure 4A). The gradient boosting (GB) model trained on the 22-feature signature selected using the p.adjust achieved predictive performance equivalent to that of the GB model trained on the full feature set (214 features) without FS (AUC = 0.71). Note that the strong performance of the no-FS GB model can be attributed to the restriction of the depth of tree-based classifiers in the benchmark to 2, which added an additional internal layer of FS.

Overall, most benchmarked methods achieved optimism-corrected AUC values comparable to standard LASSO, although several approaches, predominantly statistical filters, showed modest improvements. However, more than half of the methods demonstrated higher stability than LASSO, supporting the idea that it is possible to select features more consistently while maintaining similar predictive performance [41]. Across all methods, the association between stability and AUC was weak (Spearman's $r = 0.32$, $p = 0.044$). The least stable approaches tended to perform poorly, but high stability did not necessarily translate into top predictive performances.

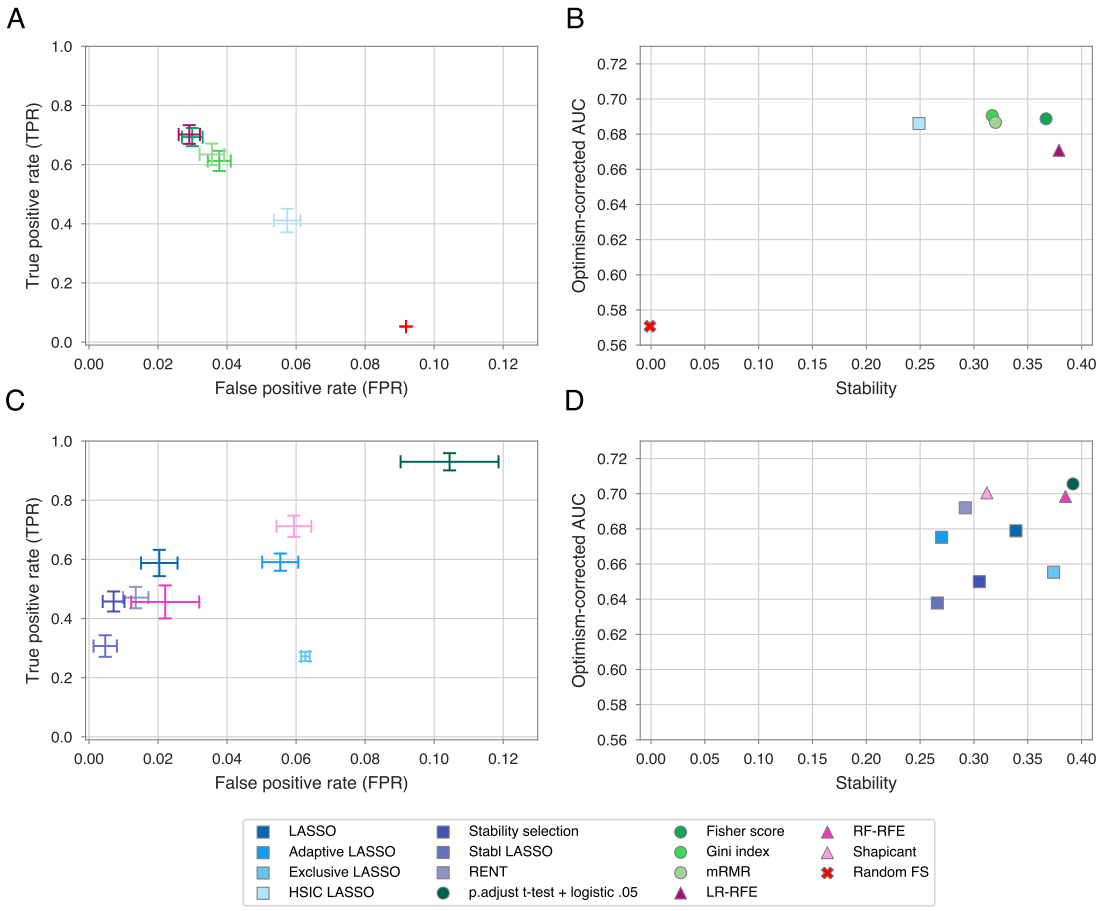


Figure 4: Performance of representative FS methods from different algorithmic families (top: FS methods with fixed, user-defined subset size; bottom: FS methods with subset size based on algorithm thresholds). Blue: embedded methods; violet-blue: resampling-based ensemble methods; green: filters; rose: wrappers. **A, C:** Trade-off between discovery and error in the experiment on the semi-synthetic PIONeeR dataset with a simulated outcome. The true positive rate denotes the proportion of true predictors correctly identified by each FS method, while the false positive rate shows the proportion of non-informative features incorrectly selected. Error bars represent one-half of the standard deviation (SD). **B, D:** Stability and predictive performance (AUC) in the experiment on the real PIONeeR dataset.

3.4 Benchmarking on semi-synthetic data shows two distinct patterns in the relationship between true and false discoveries

The second part of the benchmark was conducted on a semi-synthetic dataset to assess the ability of FS methods to discover true predictors. As detailed in Methods, a synthetic outcome variable was simulated with a simple linear model from 19 pre-selected features from the real PIONeeR dataset, and FS methods were applied to the original dataset combined with this simulated outcome. Their performance was measured using true positive rate (TPR), false positive rate (FPR), false discovery rate (FDR) and false omission rate (FOR).

Even though the outcome was generated with a simple linear model, no method achieved both high TPR and low FPR (Table 2, fig. 4B). By construction, fixed-size methods exhibited an inverse relationship between TPR and FPR, as true discoveries constrained the number of false positives (fig. 4). In contrast, threshold-based methods displayed the opposite behavior: methods with higher TPR also had higher FPR. More relaxed methods (e.g., LASSO) showed higher TP and FP rates, whereas more stringent methods (e.g., Stable LASSO) produced smaller TP and FP rates. The same behavior was observed when FPs were represented as the false discovery rate (FDR), i.e., the proportion of FPs relative to the signature size (supplementary fig. S1).

As expected, embedded ensemble methods that control for false discoveries (stability selection, Stabl LASSO, RENT) achieved the lowest FPR but at the cost of lower TPR. These methods are characterized by consistent strict selection of features with the strongest signal. However, they omit a high proportion of true predictors with weaker signals, demonstrating the TPR of only 0.46 ± 0.07, 0.31 ± 0.07 and 0.47 ± 0.07, respectively. Statistical filters exhibited the highest TPR values, while their FPR remained moderate, with one exception. p.adjust filter achieved the highest TPR (0.93 ± 0.06), rarely omitting true predictors, but at the cost of high FPR and FDR, despite BH adjustment being explicitly designed to control false discoveries. FPR of p.adjust was significantly higher than that of random FS (0.1 ± 0.03 vs 0.07 ± 0.03; Mann–Whitney U test, $p < 0.0001$), while its FDR was high but significantly lower than that of random FS (0.53 ± 0.06 vs 0.89 ± 0.04; Mann–Whitney U test, $p < 0.0001$). The high FPR of p.adjust may be explained by residual correlations among predictors in the post-VIF dataset, which can inflate false discoveries in multiple hypothesis testing settings, as demonstrated by Kanduri et al. [43]. Finally, information theory filters performed poorly, exhibiting both high FPR and low TPR, consistent with their omission of known biomarkers in the real outcome analysis.

Overall, Table 2 illustrates that a low FPR did not necessarily correspond to better predictive performance, whereas higher TPR values were generally associated with improved AUC. This highlights that generalisation requires identifying a sufficient set of true predictors rather than minimising false positives alone.

Table 2 additionally compares computational efficiency across FS methods. Most filter and embedded methods required ≤ 10 seconds per bootstrap iteration (and ≤ 1 second for some). In contrast, wrapper methods and resampling-based approaches were substantially more demanding, with computation time approaching 1 minute per bootstrap iteration.

3.5 p.adjust filter achieves a more stable and predictive signature

Based on the results presented above, the p.adjust filter based on the union of BH-adjusted p-values from logistic regression (controlled for PD-L1) and t-test with 0.05 threshold was identified as the top-performing method for predicting the resistance to immunotherapy in the PIONeeR data. It demonstrated the highest optimism-corrected AUC (0.72) when fitted with the gradient boosting classifier. It also ranked among the most stable methods (S = 0.39). When benchmarked on semi-synthetic dataset, it achieved the highest TPR of 0.93 ± 0.06 at the expense of a quite high FPR of 0.1 ± 0.03. This was acceptable in our setting, as the cost of missing a true biomarker was considered to be higher than that of including redundant biomarkers. Importantly, unlike standard LASSO and Shapicant, a representative of wrapper methods, all p.adjust-derived features were selected in $> 50\%$ of bootstrap samples, increasing the reliability of the resulting signature (Figure 5A).

Additionally, we evaluated the stability of the p.adjust models at the individual patient level as measured by the instability index [44] – the proportion of bootstrap models that classify a given patient differently compared to the full model (Figure 5B). Ideally, the instability index should be elevated only for patients whose predicted probabilities lie close to the decision threshold of the full model. All models lacked this behavior. However, overall p.adjust improved patient-level stability: only 23.4% of patients had an instability index exceeding 0.1, compared to 29.9% for Shapicant and 32.0% for LASSO. At the more strict threshold of 0.3, this proportion decreased to 9.4% of patients for p.adjust, versus 12.4% for Shapicant and 9.9% for LASSO.

Although the signature selected by the p.adjust on the full dataset comprised 22 features, bootstrap-derived subsets included up to 72 features (Figure 5C). We investigated this by repeating the experiment using subsampling without replacement at 80% of the original sample size. Under subsampling, the inflation of selected subset size disappeared, while stability further increased and predictive performance was preserved. This suggests that the larger subset sizes observed with bootstrapping may result from over-representation of specific patient subclusters, increasing sample-specific signals that are weak in the full dataset. Therefore the sparsity instability of p.adjust appears to result from the bootstrapping procedure rather than the method itself.

In conclusion, conducting the FS benchmark using *roofs* and comparing the results obtained by multiple FS methods allowed us to improve performance and reliability from the baseline FS approach by identifying p.adjust as a method with optimal stability and AUC on the data with the real outcome and the best balance between TPR and FPR on the data with the simulated outcome.

4 Discussion

This paper introduced *roofs*, a Python framework designed to help researchers identify an FS method adapted to their predictive task. FS is known to be data-dependent, with method performance varying across datasets and no single approach being universally optimal [45, 19, 46]. However, FS methods are often adopted from the literature or research teams established practices, with little consideration of their suitability for the specific dataset at hand. *roofs* addresses

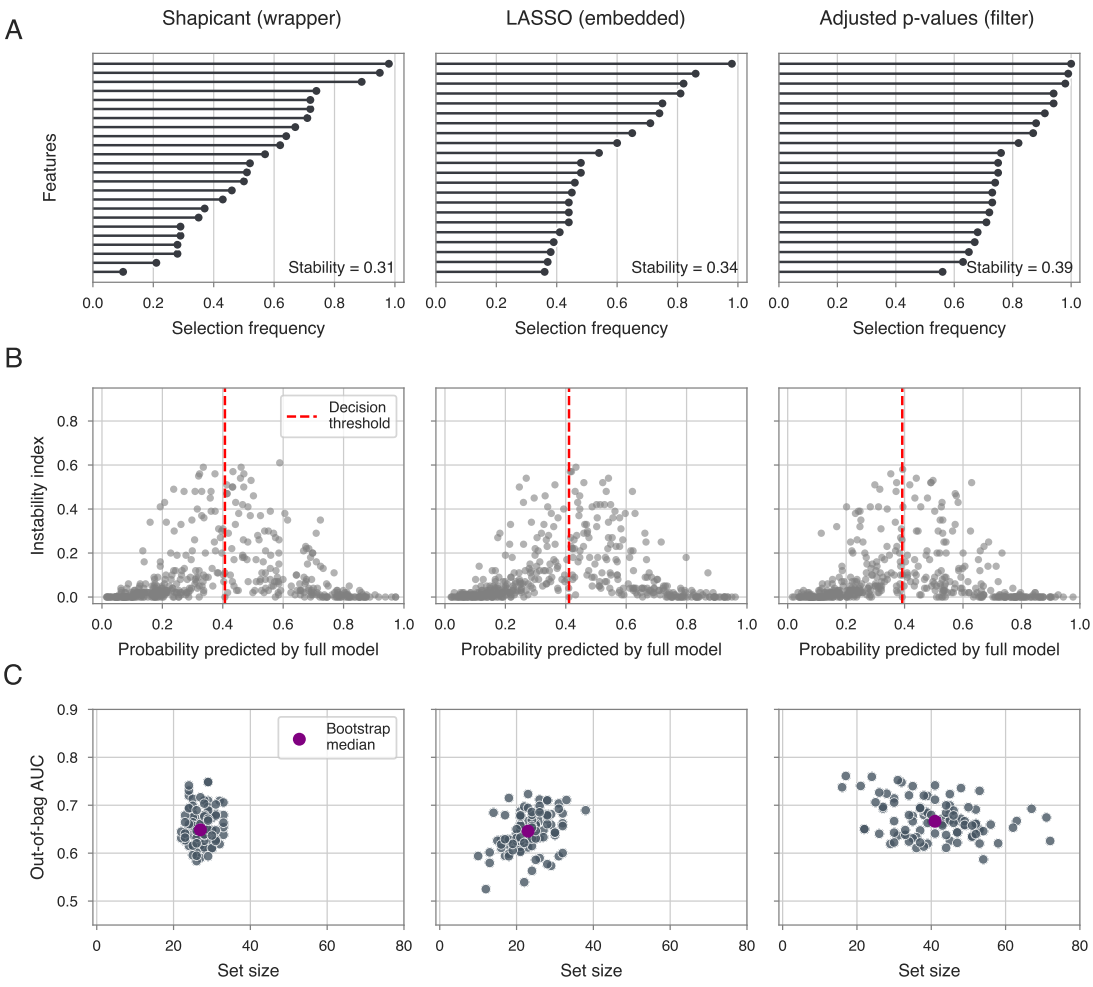


Figure 5: Comparison of performance between Shapicant (representative wrapper method), LASSO (baseline embedded method), and p.adjust (the FS approach selected using *roofs*). **A:** Selection frequency of signature features. **B:** Instability index, defined as the proportion of bootstrap models yielding a different classification for a given patient compared to the full model [44]. **C:** Out-of-bag AUC (θ_{OOB}; see Methods) as a function of selected subset size across bootstrap models.

this gap by integrating the evaluation of over 30 FS methods from different algorithmic families into a single pipeline that requires low to zero additional programming. It generates automatic reports summarizing FS performance metrics, as well as measures of individual feature robustness. A key element of *roofs* is its emphasis on stability – a property of FS that is frequently ignored in biomedical studies and in the development of new FS approaches despite its importance for reproducibility.

To our knowledge, no previously released packages combines multiple pre-implemented FS methods with the optimism correction framework for model evaluation. The closest existing tool is the R package *pminternal* (https://github.com/stephenrho/pminternal), which estimates optimism for user-defined model functions, including those that contain an FS step. However, it does not provide any pre-implemented FS methods.

Currently, *roofs* is limited to classification tasks. Future work will focus on extending it to regression and survival tasks. Another limitation is that *roofs* requires considerable runtime to perform a full benchmark. While all implemented methods are scalable to moderately high-dimensional data, in datasets with thousands of features, it may be reasonable to reduce the runtime by avoiding wrapper and ensemble methods and limiting the number of classifiers used in downstream prediction.

We demonstrated the utility of *roofs* through application to the dataset from the PIONeeR clinical trial, focused on predicting resistance to anti-PD-(L)1 immunotherapy in advanced lung cancer. It enabled identification of the adjusted p-values filter – the union of a t-test and logistic regression controlled for PD-L1 – as an FS method offering the best stability, confidence in selected features, predictive performance and balance between TPR and FPR. The general findings of this study align with previous benchmarks where simple statistical filters outperformed more complicated ML approaches [2]. Therefore, we recommend starting the analysis of new datasets by benchmarking these methods, given their additional advantage of computational efficiency.

To conclude, the choice of an FS method and, therefore, the resulting subset of features or "signature", should be driven by the dataset's characteristics and study goals. It should consider study-specific acceptable trade-off between model stability, performance, and relative costs of false positives and false negatives. *roofs* can serve as a practical tool for researchers working in settings where reproducibility is crucial. By enabling comprehensive benchmarking, *roofs* and similar tools have the potential to improve the robustness and reproducibility of biomarker discovery and increase the translational value of clinical predictive models.

Funding statement

This work benefited from a government grant handled by the French National Research Agency (ANR) as part of the France 2030 investment plan, under the reference ANR-17-RHUS-0007.

This work was supported by a partnership of Aix-Marseille Université (AMU), Assistance Publique Hôpitaux de Marseille (APHM), Centre National de La Recherche Scientifique (CNRS), Institut National de la Santé et de la Recherche Médicale (INSERM), Centre Léon Bérard (CLB), Institut Paoli Calmettes (IPC), Gustave Roussy (GR), AstraZeneca (AZ), Veracyte (VERA), Innate Pharma (IPH) & ImCheck Therapeutics (ICT), and initiated by Marseille Immunopole.

The authors gratefully acknowledge the support of the APHM, which sponsored the PIONeeR clinical studies. Its role was to control the appropriateness of ethical and legal considerations for all centers and to perform the monitoring of the consents signed and the clinical data recorded and coded as part of the study. The authors are grateful to all the patients and their families, as well as all the investigators, for their participation in the study.

This work was also supported by a grant from the French government, managed by the National Research Agency (ANR), under the France 2030 program within the DIGPHAT project, reference ANR-22-PESN-0017.

References

[1] Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, page 218–225, USA, 2005. IEEE Computer Society. ISBN 0769522785. doi: 10.1109/ICDM.2005.135. URL `https://doi.org/10.1109/ICDM.2005.135`.

[2] Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLOS ONE*, 6(12):e28210, 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0028210.

[3] Georg Heinze, Christine Wallisch, and Daniela Dunkler. Variable selection – a review and recommendations for the practicing statistician. *Biometrical Journal*, 60(3):431–449, 2018. ISSN 1521-4036. doi: 10.1002/bimj.201700067.

[4] Borja Seijo-Pardo, Amparo Alonso-Betanzos, Kristin P. Bennett, Verónica Bolón-Canedo, Julie Josse, Mehreen Saeed, and Isabelle Guyon. Biases in feature selection with missing data. *Neurocomputing*, 342:97–112, May 2019. ISSN 0925-2312. doi: 10.1016/j.neucom.2018.10.085.

[5] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys*, 50(6):94:1–94:45, 2017. ISSN 0360-0300. doi: 10.1145/3136625.

[6] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143:106839, 2020. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2019.106839. URL `https://www.sciencedirect.com/science/article/pii/S016794731930194X`.

[7] Aydin Demircioğlu. Benchmarking feature projection methods in radiomics. *Scientific Reports*, 15(1):32368, sept 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-16070-w.

[8] Shaveta Tatwani and Ela Kumar. Effect of subset size on the stability of feature selection for gene expression data. In *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2019*, pages 541–546, London, U.K., July 2019.

[9] Gary S Collins, Paula Dhiman, Jie Ma, Michael M Schlussel, Lucinda Archer, Ben Van Calster, Frank E Harrell, Glen P Martin, Karel G M Moons, Maarten van Smeden, Matthew Sperrin, Garrett S Bullock, and Richard D Riley. Evaluation of clinical prediction models (part 1): from development to external validation. *BMJ*, 384, 2024. doi: 10.1136/bmj-2023-074819. URL https://www.bmj.com/content/384/bmj-2023-074819.

[10] Konstantin Hopf and Sascha Reifenrath. Filter methods for feature selection in supervised machine learning applications - review and benchmark. *ArXiv*, abs/2111.12140, 2021. URL https://api.semanticscholar.org/CorpusID:244527455.

[11] Luca Cattelani, Arindam Ghosh, Teemu J. Rintala, and Vittorio Fortino. A comprehensive evaluation framework for benchmarking multi-objective feature selection in omics-based biomarker discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 21(6):2432–2446, November 2024. ISSN 1557-9964. doi: 10.1109/TCBB.2024.3480150.

[12] Justine Labory, Evariste Njomgue-Fotso, and Silvia Bottini. Benchmarking feature selection and feature extraction methods to improve the performances of machine-learning algorithms for patient classification using metabolomics biomedical data. *Computational and Structural Biotechnology Journal*, 23:1274–1287, 2024. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2024.03.016. URL https://www.sciencedirect.com/science/article/pii/S2001037024000692.

[13] Sugam Budhraja, Maryam Doborjeh, Balkaran Singh, Samuel Tan, Zohreh Doborjeh, Edmund Lai, Alexander Merkin, Jimmy Lee, Wilson Goh, and Nikola Kasabov. Filter and wrapper stacking ensemble (fwse): a robust approach for reliable biomarker discovery in high-dimensional omics data. *Briefings in Bioinformatics*, 24(6): bbad382, September 2023. ISSN 1477-4054. doi: 10.1093/bib/bbad382.

[14] Yingxia Li, Ulrich Mansmann, Shangming Du, and Roman Hornung. Benchmark study of feature selection strategies for multi-omics data. *BMC Bioinformatics*, 23(1):412, October 2022. ISSN 1471-2105. doi: 10.1186/s12859-022-04962-x.

[15] Andrea Bommert, Thomas Welchowski, Matthias Schmid, and Jörg Rahnenführer. Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Briefings in Bioinformatics*, 23(1):bbab354, January 2022. ISSN 1477-4054. doi: 10.1093/bib/bbab354.

[16] Tapas Bhadra, Saurav Mallik, Neaj Hasan, and Zhongming Zhao. Comparison of five supervised feature selection algorithms leading to top features and gene signatures from multi-omics data in cancer. *BMC Bioinformatics*, 23 (3):153, April 2022. ISSN 1471-2105. doi: 10.1186/s12859-022-04678-y.

[17] Chih-Wen Chen, Yi-Hong Tsai, Fang-Rong Chang, and Wei-Chao Lin. Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Systems*, 37(5):e12553, October 2020. ISSN 0266-4720. doi: 10.1111/exsy.12553.

[18] Naiyer Rizvi, Foluso O Ademuyiwa, Z Alexander Cao, Helen X Chen, Robert L Ferris, Sarah B Goldberg, Matthew D Hellmann, Ranee Mehra, Ina Rhee, Jong Chul Park, Harriet Kluger, Hussein Tawbi, and Ryan J Sullivan. Society for immunotherapy of cancer (sitc) consensus definitions for resistance to combinations of immune checkpoint inhibitors with chemotherapy. *Journal for Immunotherapy of Cancer*, 11(3):e005920, March 2023. ISSN 2051-1426. doi: 10.1136/jitc-2022-005920.

[19] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Publishing Company, Incorporated, 2017. ISBN 978-1-4614-7137-0.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Julien Hédou, Ivana Marić, Grégoire Bellan, Jakob Einhaus, Dyani K. Gaudillière, Francois-Xavier Ladant, Franck Verdonk, Ina A. Stelzer, Dorien Feyaerts, Amy S. Tsai, Edward A. Ganio, Maximilian Sabayev, Joshua Gillard, Jonas Amar, Amelie Cambriel, Tomiko T. Oskotsky, Alennie Roldan, Jonathan L. Golob, Marina Sirota, Thomas A. Bonham, Masaki Sato, Maïgane Diop, Xavier Durand, Martin S. Angst, David K. Stevenson, Nima Aghaeepour, Andrea Montanari, and Brice Gaudillière. Discovery of sparse, reliable omic biomarkers with stabl. *Nature Biotechnology*, January 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-02033-x. URL https://doi.org/10.1038/s41587-023-02033-x.

[22] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246.

[23] Francis R. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 33–40, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390161. URL https://doi.org/10.1145/1390156.1390161. event-place: Helsinki, Finland.

[24] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476): 1418–1429, December 2006. ISSN 0162-1459. doi: 10.1198/016214506000000735.

[25] Yang Zhou, Rong Jin, and Steven Chu-Hong Hoi. Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, page 988–995. JMLR Workshop and Conference Proceedings, March 2010. URL `https://proceedings.mlr.press/v9/zhou10a.html`.

[26] Makoto Yamada, Jiliang Tang, Jose Lugo-Martinez, Ermin Hodzic, Raunak Shrestha, Avishek Saha, Hua Ouyang, Dawei Yin, Hiroshi Mamitsuka, Cenk Sahinalp, Predrag Radivojac, Filippo Menczer, and Yi Chang. Ultra high-dimensional nonlinear feature selection for big biological data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1352–1365, July 2018. ISSN 1558-2191. doi: 10.1109/TKDE.2018.2789451.

[27] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2010.00740.x.

[28] Anna Jenul, Stefan Schrunner, Kristian Hovde Liland, Ulf Geir Indahl, Cecilia Marie Futsæther, and Oliver Tomic. Rent—repeated elastic net technique for feature selection. *IEEE Access*, 9:152333–152346, 2021. doi: 10.1109/ACCESS.2021.3126429.

[29] Dahua Lin and Xiaoou Tang. Conditional infomax learning: An integrated framework for feature extraction and fusion. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 68–82, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33833-8.

[30] François Fleuret. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.*, 5: 1531–1555, December 2004. ISSN 1532-4435.

[31] Patrick E. Meyer and Gianluca Bontempi. On the use of variable complementarity for feature selection in cancer classification. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, Ernesto Costa, Carlos Cotta, Rolf Drechsler, Evelyne Lutton, Penousal Machado, Jason H. Moore, Juan Romero, George D. Smith, Giovanni Squillero, and Hideyuki Takagi, editors, *Applications of Evolutionary Computing*, pages 91–102, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33238-1.

[32] Howard Yang and John Moody. Data visualization and feature selection: New algorithms for nongaussian data. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL `https://proceedings.neurips.cc/paper_files/paper/1999/file/8c01a75941549a705cf7275e41b21f0d-Paper.pdf`.

[33] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1):23–69, October 2003. ISSN 1573-0565. doi: 10.1023/A:1025667309714.

[34] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 0035-9246.

[35] Manuel Calzolari. Shapicant: feature selection package based on shap and target permutation. `https://github.com/manuel-calzolari/shapicant`, 2020. Version 0.4.0.

[36] F. E. Harrell, K. L. Lee, and D. B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, 15(4):361–387, February 1996. ISSN 0277-6715. doi: 10.1002/(SICI)1097-0258(19960229)15:4%3C361::AID-SIM168%3E3.0.CO;2-4.

[37] Bradley Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983. ISSN 0162-1459. doi: 10.2307/2288636.

[38] Bradley Efron and Robert Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. ISSN 0162-1459. doi: 10.2307/2965703.

[39] Michael R. Chernick and Robert A. LaBudde. *An Introduction to Bootstrap Methods with Applications to R*. Wiley Publishing, 1st edition, October 2011. ISBN 978-0-470-46704-6.

[40] Katsuhiro Iba, Tomohiro Shinozaki, Kazushi Maruo, and Hisashi Noma. Re-evaluation of the comparative effectiveness of bootstrap-based optimism correction methods in the development of multivariable clinical prediction models. *BMC Medical Research Methodology*, 21(1):9, January 2021. ISSN 1471-2288. doi: 10.1186/s12874-020-01201-w.

[41] Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the stability of feature selection algorithms. *Journal of Machine Learning Research*, 18(174):1–54, 2018. ISSN 1533-7928.

[42] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7(90): 2541–2563, 2006. ISSN 1533-7928.

[43] Chakravarthi Kanduri, Maria Mamica, Emilie Willoch Olstad, Manuela Zucknick, Jingyi Jessica Li, and Geir Kjetil Sandve. Beware of counter-intuitive levels of false discoveries in datasets with strong intra-correlations. *Genome Biology*, 26(1):249, August 2025. ISSN 1474-760X. doi: 10.1186/s13059-025-03734-z.

[44] Richard D. Riley and Gary S. Collins. Stability of clinical prediction models developed using statistical or machine learning methods. *Biometrical Journal*, 65(8):2200302, 2023. ISSN 1521-4036. doi: 10.1002/bimj.202200302.

[45] Petr Somol, Jana Novovicova, and Pavel Pudil. *Efficient Feature Subset Selection and Subset Size Optimization*. IntechOpen, February 2010. ISBN 978-953-7619-90-9. doi: 10.5772/9356. URL `https://www.intechopen.com/chapters/10666`.

[46] P. Drotár, J. Gazda, and Z. Smékal. An experimental comparison of feature selection methods on two-class biomedical datasets. *Computers in Biology and Medicine*, 66:1–10, November 2015. ISSN 0010-4825. doi: 10.1016/j.compbiomed.2015.08.010.

[47] Héctor Climente-González, Chloé-Agathe Azencott, Samuel Kaski, and Makoto Yamada. Block hsic lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics (Oxford, England)*, 35(14): i427–i435, july 2019. ISSN 1367-4811 1367-4803. doi: 10.1093/bioinformatics/btz333.

[48] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, January 2002. doi: 10.1023/A:1012487302797.

[49] Hongying Jiang, Youping Deng, Huann-Sheng Chen, Lin Tao, Qiuying Sha, Jun Chen, Chung-Jui Tsai, and Shuanglin Zhang. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5(1):81, june 2004. ISSN 1471-2105. doi: 10.1186/1471-2105-5-81.

[50] Siwei Xia and Yuehan Yang. An iterative model-free feature screening procedure: Forward recursive selection. *Knowledge-Based Systems*, 246:108745, june 2022. ISSN 0950-7051. doi: 10.1016/j.knosys.2022.108745.

5 Supplementary

Supplementary table S1: Benchmarked FS methods

Method	Description	Reference
	Embedded methods	
LASSO	Logistic regression with the L1 penalty.	[22]
Bolasso	Modified bootstrap LASSO. Selects features with a bootstrap frequency of selection of ≥ 0.5 instead of 1.	[23]
Adaptive LASSO	LASSO variant that applies adaptive weights to each coefficient, allowing to penalize unimportant variables more than important ones.	[24]
Exclusive LASSO	Applies the L1 penalty at group level: $[\arg\min_\beta \frac{1}{2n}\lvert y - X\beta\rvert_2^2 + \lambda \sum g \in \mathcal{G} \frac{\lvert\beta_g\rvert_1^2}{2}]$ In *roofs*, groups $\mathcal{G}$ are defined via hierarchical clustering, with with $\mid \mathcal{G} \mid = 0.1 \cdot p$ (p is the total number of features in the data).	[25]
Block HSIC LASSO	Selects features based on their dependence with the outcome using the Hilbert-Schmidt Independence Criterion.	[47]
	Ensemble embedded methods	
Stability selection	LASSO-based subsampling method that performs repeated fits across a range of regularization parameters and selects features selected with high frequency above a specified threshold.	[27]
Stabl LASSO	Recent extension of stability selection that injects artificial noise variables into the data to enable data-driven choice of the selection frequency threshold, with the goal of minimizing false discoveries.	[21]
RENT	Repeated Elastic Net Technique, a resampling-based method that selects features that satisfy three criteria: a sufficiently high selection frequency across resampled models, consistent coefficient sign across models, and a sufficiently large ratio of the mean coefficient to its standard error.	[28]
	Filter methods	
CIFE	Iteratively selects features with the highest Conditional Infomax Feature Extraction (CIFE) score until the user-specified set size is reached. For each feature X_k: CIFE$(X_k) = I(X_k;Y) - \sum_{X_j \in \mathcal{S}} I(X_j;X_k) + \sum_{X_j \in \mathcal{S}} I(X_j;X_k \mid Y)$ The first term maximizes mutual information (MI) between the feature and the outcome, the second penalizes redundancy with previously selected features, and the third rewards conditional redundancy given the outcome.	[5, 29]
CMIM	Iteratively selects features with the highest Conditional Mutual Information Maximization (CMIM) score until the user-specified set size is reached. For each feature X_k: CMIM$(X_k) = \min_{X_j \in \mathcal{S}}[I(X_k;Y \mid X_j)]$ The CMIM score maximizes the minimal conditional MI between the candidate feature X_k and the outcome Y, with the minimum value chosen across all previously selected features $X_j \in \mathcal{S}$.	[5, 30]
DISR	Iteratively selects features with the highest Double Input Symmetrical Relevance (DISR) score until the user-specified set size is reached. For each feature X_k: DISR$(X_k) = \sum_{X_j \in F} \frac{I(X_j X_k;Y)}{H(X_j X_k Y)}$, where $I(X_j X_k;Y) = I(X_k;Y) + I(X_j;Y \mid X_k)$, and $H(X_j X_k Y) = H(X_k) + H(X_k \mid X_j) + H(Y \mid X_k) - I(Y;X_j \mid X_k)$.	[5, 31]
JMI	Iteratively selects features with the highest Joint Mutual Information (JMI) score until the user-specified set size is reached. For each feature X_k: JMI$(X_k) = I(X_k;Y) - \beta \sum_{X_j \in \mathcal{S}} I(X_j;X_k) + \gamma \sum_{X_j \in \mathcal{S}} I(X_j;X_k \mid Y)$, where $\beta = \gamma = \frac{1}{\lvert\mathcal{S}\rvert}$, where $\mid \mathcal{S} \mid$ is the cardinality of a set to be selected.	[5, 32]
ReliefF	Ranks features by a distance-based score and selects top-k features. For each sample, the algorithm penalizes the features with different values in the 5 nearest neighbors of the same class and rewards the features with different values for the 5 nearest neighbors of the opposite class.	[5, 33]

Continued on next page

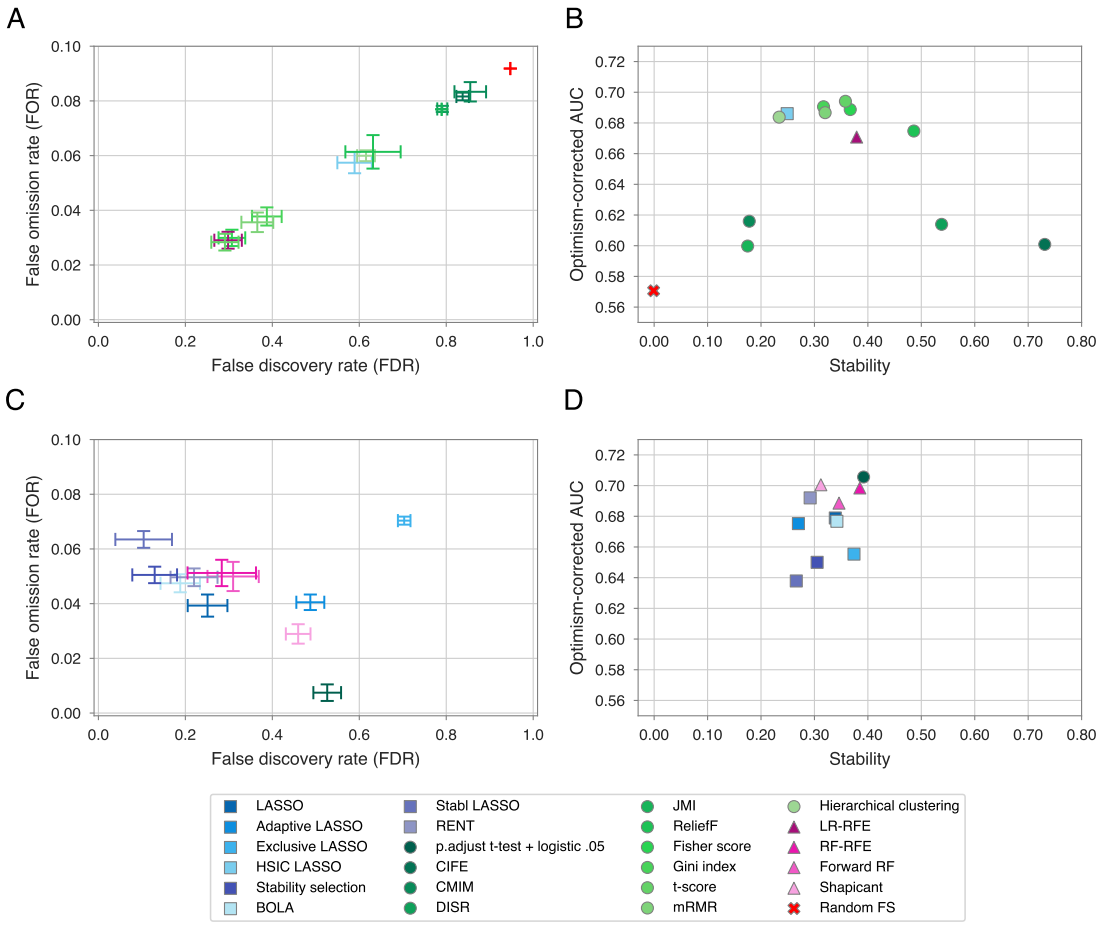
Supplementary table S1 – continued from previous page

Method	Description	Reference
Fisher score	Ranks features by the ratio of between-class variance to within-class variance and selects top-k features.	[5]
Gini index	Ranks features by their minimum Gini impurity (computed by testing all possible binary splits) and selects top-k features.	[5]
t-score	Ranks features by their two-sample t-statistic and selects top-k features.	[5]
mRMR	Iteratively adds features with the highest relevance-redundancy ratio until the desired number of features is reached. Relevance is measured by the ANOVA F-value, redundancy is iteratively updated as the mean correlation with previously selected features.	
Hierarchical clustering	Performs hierarchical clustering of features based on Spearman correlation coefficients, with the number of clusters equal to the desired number of features to be selected. From each cluster, the feature with the highest point biserial correlation with the outcome is selected.	
p.adjust t-test + logistic .05	Union-based filter that combines features selected by t-test and logistic regression (controlling for user-specified covariates; PD-L1 in the case of PIONeeR data), where p-values from each test were adjusted using the Benjamini-Hochberg procedure with a 0.05 threshold.	
Wrapper methods		
LR-RFE	Recursive feature elimination that iteratively removes 10% of features with the lowest coefficients from a logistic regression model with the L2 penalty until the pre-specified number of features is reached.	[48]
RF-RFE	Recursive feature elimination that iteratively removes 10% of the least important feature of a random forest model and selects a subset with the best AUC in cross-validation.	[49]
Forward RF	Recursive forward selection that iteratively adds the most important feature of a random forest model until a pre-specified set size is reached and selects a subset with the best AUC in cross-validation.	[50]
Shapicant	Wrapper that uses SHAP importances together with outcome permutation. The criterion of selection is the threshold on p-values from testing whether each feature's SHAP value under the real outcome differs from that under the permuted outcome. In *roofs*, the default procedure begins with a threshold of 0.1 and iteratively decreases it, stopping after three iterations or once the number of selected features k falls below the user-specified max_num_selected.	[35]
Controls		
Random FS	Selects a feature subset randomly. For the benchmark on PIONeeR data, subset sizes k were randomly sampled from the range 1-20 for the real dataset and set to $k = 19$ for the semi-synthetic dataset, corresponding to the number of predictors in the true model.	
Full model	Model trained with all features.	

Supplementary table S2: ML classifiers used in FS benchmark on PIONeeR dataset

Category	Classifier	Hyperparameters[a]
Tree-based	BaggingClassifier	`base_estimator`=DecisionTree, `n_estimators`=200 `max_depth`=2, `min_samples_leaf`=40, `max_features`=0.7
	ExtraTreesClassifier	`n_estimators`=200, `max_depth`=2 `min_samples_leaf`=40, `max_features`=0.7
	RandomForestClassifier	`n_estimators`=200, `max_depth`=2 `min_samples_leaf`=40, `max_features`=0.7
Boosting	GradientBoostingClassifier	`max_depth`=2, `min_samples_leaf`=40 `max_features`=0.7, `subsample`=0.7
	HistGradientBoostingClassifier	`max_depth`=2, `min_samples_leaf`=40, `max_features`=0.7
	XGBClassifier	`max_depth`=2, `n_jobs`=-1, other parameters at default
Linear	LogisticRegression	`penalty`=None, `n_jobs`=-1
	LinearDiscriminantAnalysis	Default hyperparameters
Other	KNeighborsClassifier	Default
	MLPClassifier	Default
	SVC	`probability`=True, other parameters at default
Ensemble	StackingClassifier[b]	`cv`=5, `stack_method=predict_proba` Base estimators: LogisticRegression, RandomForest, SVC, GradientBoosting Final estimator: LogisticRegression

[a] All classifiers use scikit-learn default parameters unless specified.
[b] StackingClassifier's base estimators use the same hyperparameters as the individual classifiers.



Supplementary figure S1: Performance of all benchmarked FS methods (top: FS methods with fixed, user-defined subset size; bottom: FS methods with subset size based on algorithm thresholds). Blue: embedded methods; violet-blue: resampling-based ensemble methods; green: filters; rose: wrappers.**A, C:** False omission and false discovery rates in the experiment on the semi-synthetic PIONeeR dataset with a simulated outcome. The false omission rate represents the proportion of true features that were incorrectly not selected (false negatives) among all non-selected features, while the false discovery rate represents the proportion of incorrectly selected features (false positives) relative to the size of the signature derived by an FS algorithm. Error bars represent one-half of the standard deviation (SD). **B, D:** Stability and predictive performance (AUC) in the experiment on the real PIONeeR dataset.