Computers in Biology and Medicine 152 (2023) 106413

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

SEQENS: An ensemble method for relevant gene identification in microarray data

François Signol *, Laura Arnal, J. Ramón Navarro-Cerdán, Rafael Llobet, Joaquim Arlandis, Juan-Carlos Perez-Cortes

Instituto Tecnológico de Informática (ITI), Universitat Politècnica de València, Camino de Vera, s/n, 46022 València, Spain

ARTICLE INFO

Keywords:
Gene identification
Feature selection
Ensemble method
Microarray data
High dimensionality spaces

ABSTRACT

This paper describes an ensemble feature identification algorithm called SEQENS, and measures its capability to identify the relevant variables in a case-control study using a genetic expression microarray dataset. SEQENS uses Sequential Feature Search on multiple sample splitting to select variables showing stronger relation with the target, and a variable relevance ranking is finally produced. Although designed for feature identification, SEQENS could also serve as a basis for feature selection (classifier optimisation). Cliff, a ranking evaluation metric is also presented and used to assess the feature identification algorithms when a groundtruth of relevant variables is available. To test performance, three types of synthetic groundtruths emulating fictitious diseases are generated from ten randomly chosen variables following different target pattern distributions using the E-MTAB-3732 dataset. Several sample-to-dimensionality ratios ranging from 300 to 3,000 observations and 854 to 54,675 variables are explored. SEQENS is compared with other feature selection or identification state-of-the-art methods. On average, the proposed algorithm identifies better the relevant genes and exhibits a stronger stability. The algorithm is available to the community.

1. Introduction

Being able to identify genes related to a phenotype, or a particular disease, is one of the main challenges for artificial intelligence and biocomputing. This process is known as gene identification in genomics and is related to feature selection in machine learning. Identifying such relevant variables could lead to the discovery or confirmation of biological and medical knowledge, and foster current debates about understanding disease mechanisms. Moreover, it is an important step to be able to develop decision support tools based on machine learning useful for clinical practice.

Genetic analysis typically produces high-throughput sequencing or microarray hybridisation. In this context, the number of variables (gene expression, variants and mutations, RNA slices, protein coding, etc.) to be analysed can be vast, which may require considerable effort in terms of algorithm complexity and computational resources.

Dealing with very high dimensional data comes with important obstacles inherent to the curse of dimensionality [1]. Indeed, when the quantity of variables is much greater than the quantity of observations, the probability of detecting false relationships between one or more variables and a phenotype increases. The method proposed in this paper is intended to minimise this phenomenon.

Gene identification, or more generally feature identification (FI), is linked with feature selection (FS) as both approaches are based on ranking the variables by relevance (importance) in relation to a target (phenotype), though they differ in their purpose. Thus, feature selection, looks for the smallest subset of variables with the best predictive power, whereas feature identification attempts to identify all the variables that have an influence on the target. The subset of relevant variables has not necessarily the best predictive performance due to the possible redundancies that may exist among them.

Gene selection (or feature selection) is usually evaluated by measuring the quality of the classification (or regression) obtained by the selected subset of variables [2–4]. In that case, there is no need to know the relevant variables beforehand.

In this paper, the interest is focused on identifying *all* the relevant features related to a target. A groundtruth of these influential variables is therefore necessary, since the predictive power of a set of variables is no longer a good indicator of performance. Instead, a direct performance metric, Cliff, that explicitly uses that groundtruth information is proposed.

Physical and chemical processes taking place in living bodies originate from complex gene–gene interactions (epistasis) resulting in

* Corresponding author.
E-mail addresses: fsignol@iti.es (F. Signol), larnal@iti.es (L. Arnal), jonacer@iti.es (J.R. Navarro-Cerdán), rllobet@iti.es (R. Llobet), arlandis@iti.es (J. Arlandis), jcperez@iti.es (J.-C. Perez-Cortes).

https://doi.org/10.1016/j.compbiomed.2022.106413
Received 28 April 2022; Received in revised form 25 November 2022; Accepted 3 December 2022
Available online 6 December 2022

phenotypes [5]. While the groundtruth (relevant variables) is normally not known a priori, the solution adopted in this paper is to generate it synthetically. It allows us to model a number of more, or less, complex and controlled combinations between relevant variables. FI algorithms should be capable of discovering these dependencies. In this paper, three different methods of generating synthetic groundtruths have been used to emulate fictitious diseases.

SEQENS, an ensemble feature identification method, is presented and compared with other state-of-the-art methods in the task of identifying relevant variables. The proposed algorithm ranks features (genes) by the importance they have in relation with a target (value to predict). This is done on a real gene expression dataset, and experiments are designed to test performance when the number of variables is several tens of times greater than the number of observations. This paper focuses on case-control studies from microarrays. This involves continuous-valued data and two-class targets, representing the expression of genes of interest in a population sample with individuals having, or not, a disease.

When looking through gene expression association studies, many FI methods rely on statistical tests (Welch two-sample t-test, ANOVA, etc.) [6,7]. The limits of statistical tests are understood and range from a lack of stability (two statistical tests can give rise to two very different relevance ranking) to a high rate of false positives (noise variables considered as relevant) [8].

In the machine learning literature, methods addressing feature selection are grouped in three main categories: *filtering*, *wrapper*, and *embedded*. Several reviews have gathered and compared characteristics of methods belonging to the three categories [9–11].

Filtering approaches rely on a variety of statistics computed from the data to identify variables according to a correlation or entropy criterion against the target values [12,13]. They are usually fast to compute and does not depend on a model. However, they often consider the variables independently of each other, which can hinder the discovery of interesting interactions.

In contrast, wrapper and embedded approaches are model-dependent and use error estimation as a measure to quantify feature importance. While embedded methods incorporate feature selection as part of the prediction model creation [9], wrapper methods incorporate predictors to the feature selection process [14]. An important characteristic of embedded and wrapper methods is that they can often take into account more complex variable interactions than filtering approaches. The use of a model usually comes with a higher computational cost.

Other reviews of FS methods specifically applied to microarray datasets can be found in [15–18].

In an attempt to overcome the limitations described above, the FI algorithm proposed in this paper combines wrapper feature selectors on multiple sample splits. The mathematical model behind is similar to bootstrapping but without replacement. In each split, a part of the samples goes to train an estimator and the rest goes to test (evaluate) it. Wrapper feature selectors were chosen as they allow interactions between genes to be easily explored.

These feature selectors make use of an induction algorithm (classifier or regressor, often called *inducer*, *estimator*, *learner*, or *predictor*) to guide the selection towards the subset of variables that best predicts. An exhaustive exploration of all the possible combinations of genes rapidly becomes prohibitive as their number is increased. Fortunately, the exploration workload can be drastically reduced using heuristic and meta-heuristic search strategies, such as Sequential Feature Search (SFS) [19,20], genetic algorithms [21–23], swarm algorithms [24,25], branch-and-bound approaches [26], or random subspace selection. SEQENS makes use of Sequential Feature Search.

The choice of the inducer is an important parameter. Some of them make assumptions about the type of interactions between variables (linear models, logistic regression), others do not (k-NearestNeighbours, decision trees). In this paper, the proposed SEQENS algorithm uses a k-NearestNeighbours regressor, which allows to make no assumptions about variable relationships nature since, in general, it is not known a priori.

The ensemble paradigm has been initially used for prediction tasks [27], like in Random Forest [28] or Gradient Boosting [29] algorithms. It consists of a set of individually trained inducers whose predictions are combined [30,31]. Variety can be achieved using different types of inducers. More recently, the use of ensembles has been extended to feature selection [11].

FI could suffer from lack of stability (and, consequently, low reliability) [32], particularly when the number of observations available is low compared to the number of features [33]. Experimental studies have shown that ensembles out-perform other FS algorithms in terms of stability [10,34], which can be improved by means of data splitting or *data perturbation* [35]. This effect is particularly beneficial in the context of high-dimensional/small sample size, e.g., in genetic data domains [36–38]. Thus, the need for stable feature rankings when exploring interactions between genes makes the combination of ensemble along with wrapper feature selection, such as the one presented in this paper, a plausible strategy.

The document is articulated as follows: Section 2 describes the dataset, the groundtruth generation, the tested FI algorithms and the evaluation metric. Section 3 details algorithm configurations, describes the experiments and shows the results obtained. Section 4 presents a discussion of some relevant findings. Finally, conclusions and perspectives are drawn in Section 5.

2. Material and methods

2.1. Dataset

The publicly available gene expression database used in this work is named E-MTAB-3732.[1] It is a compiled human gene expression, ontology-annotated dataset including 27,887 Affymetrix HG-U133Plus2 samples, filtered for quality control as described in [39]. No missing values were found in the database. It contains 54,675 genes from healthy and with disease individuals. Brute gene expression are continuous values varying between 2 and 15 approximately. Each gene has been standardised to obtain a zero-mean and unit-standard-deviation variable.

2.2. Synthetic case-control groundtruth generation

A performance study of several feature identification algorithms on microarray data is presented in this paper. Given the difficulty of obtaining a dataset associated with a confident groundtruth of relevant variables, fictitious diseases are generated from a pre-defined number of variables of the E-MTAB-3732 microarray dataset. These variables will be considered as the relevant ones for the generated control-case targets.

To build the groundtruth, 10 generative (relevant) variables are randomly chosen from among the 54,675 variables available in the dataset (from now on denoted by F1 to F10). Fig. 1 shows their distributions after standardisation[2] The remaining variables will be considered as noise variables (irrelevant). The FI algorithm performances will be measured by their ability to find and locate the generative variables in the first positions of their relevance rankings.

Three types of interaction between relevant variables have been tested: (1) Linear interaction in which the variables combine so as to separate cases and controls by a hyperplane; (2) A spherical interaction where controls are grouped around a reference point and cases appear

[1] https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-3732.

[2] For the sake of experiment replication or performance comparisons, the indexes (starting at 0) of F1 to F10 are 1486, 7201, 19 287, 27 461, 28 578, 29 884, 30 555, 34 271, 37 922, and 41 109.

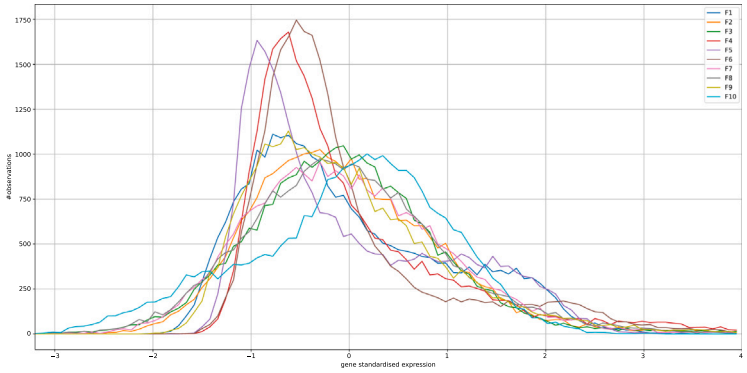

Fig. 1. Histograms of the 10 relevant variables (F1–F10) after standardisation.

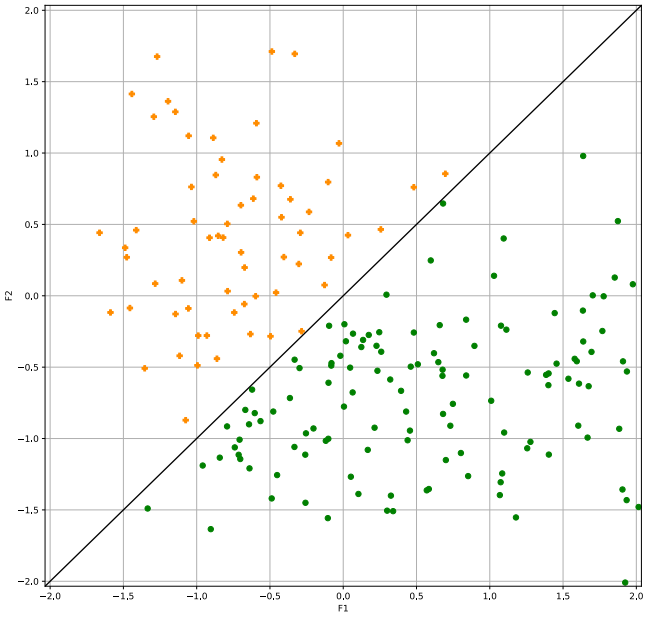

Fig. 2. Hyperplane scenario. A 2D-example built from 200 observations (patients), using two relevant variables (F1 and F2). Controls are green circles and cases are orange plus. The black line is the frontier between classes.

as one moves away from this point, hence the frontier between controls and cases is a hypersphere, and; (3) Clustered interaction where cases and controls are grouped in separate regions of the hyperspace. These scenarios are respectively called *hyperplane*, *hypersphere* and *k-Clusters*, and they are meant to be an approximation to a diversity of tasks that could be encountered in the real world, such as those using either genomic expression data, or clinical, anthropomorphic and environmental parameters. In every scenario, 1/3 of the individuals (observations) have been labelled as cases, and the remainder as controls (case-control ratio is 1/3). The generation process is detailed in the following sections.

2.2.1. Hyperplane scenario

A linear interaction separates cases and controls with a hyperplane in the subspace defined by the relevant variables. To illustrate the result of the target generation, a two-dimensional example using the (F1,F2) variable subspace is presented in Fig. 2 (200 patients from E-MTAB-3732 dataset are shown). This scenario has been inspired by the computation of a Genetic Risk Score (GRS) as proposed in [40].

Target labelling is determined by the formula of a D-dimensional hyperplane given in Eq. (1), where D is the number of relevant variables, w_d is the coefficient associated to the variable v_d. C has been set to 0 causing the hyperplane to pass through the origin point. The coefficients w_d have been randomly set to -1 or 1 producing a hyperplane with a 45 degrees slope in every dimension, giving this way the same relevance to each variable.

$$\sum_{d=1}^{D} w_d v_d + \mathrm{C} = 0 \tag{1}$$

This scenario emulates the presence (or absence) of a disease depending on whether the relevant variables have values more frequently within the first or the second half of their range.

2.2.2. Hypersphere scenario

This scenario allows us to explore a non-linear interaction. Controls are located close to the origin point and cases appear beyond a certain distance (radius). The resulting frontier between controls and cases becomes a hypersphere. The radius of the hypersphere is computed so that the case-control ratio is 1/3. Fig. 3 shows a two-dimensional example.

This scenario emulates the presence (or absence) of a disease depending on whether the relevant variables have values more frequently within a subrange or in both ends of their range.

2.2.3. k-clusters scenario

The distance that separates genotyping from phenotyping is covered by the processes of transcription and translation followed by the protein

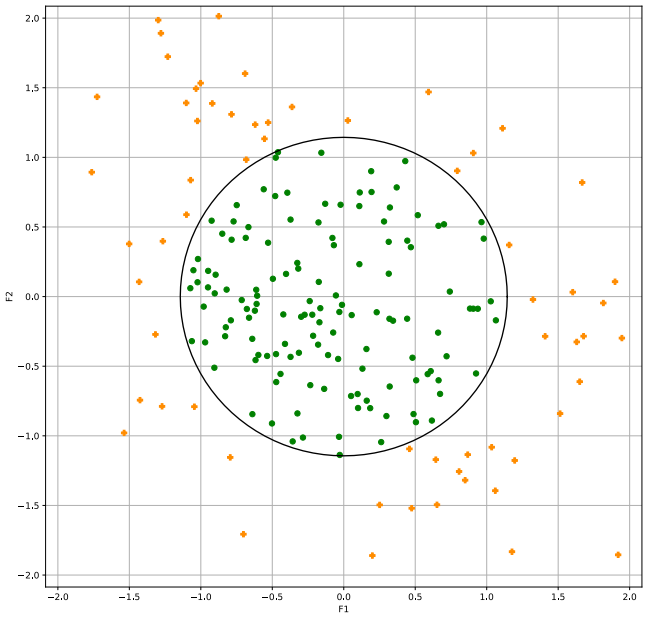

Fig. 3. Hypersphere scenario. A 2D-example built from 200 observations (patients) using two relevant variables (F1 and F2). Controls are green circles and cases are orange plus.

interactions in biological pathways. This additional complexity is simulated through a non-linear geometric scenario: the cause–effect between gene expression and a disease is modelled in a multi-dimensional space by grouping observations in a number of clusters resulting from multiple variable interactions.

Thus, observations are grouped in k clusters by means of a k-means algorithm[3] in the subspace of the relevant variables. In this work, a number of 16 clusters ($k = 16$) is chosen for experimentation. Half of the clusters are randomly assigned to controls and the other half to cases. To reach a 1/3 case-control ratio, the quantity of observations inside a case cluster is a half that in a control cluster.

As an illustrative example, two hundred patients are represented in the (F1,F2) relevant variables subspace. It is displayed as a Voronoi diagram in Fig. 4, where each background colour corresponds to a cluster. Orange plus and green circles denote cases and controls, respectively, as black triangles are the sixteen cluster centroids.

In this scenario, relevant variables can contribute to protecting from diseases in certain regions of the hyperspace while promoting them in other regions. This scenario can be considered as more complex regarding the interaction patterns among variables.

2.3. SEQENS: an ensemble feature identification method

SEQENS is an ensemble feature identification method whose kernel is a Sequential Feature Search (SFS) algorithm [20]. As a wrapper method, SFS benefits SEQENS by identifying potential interactions among relevant features using inducers as selectors, even with high dimensional spaces. The components of the proposed method are depicted in Fig. 5 and detailed in the following subsections.

The fundamentals of ensemble methods for feature selection are to combine the results of multiple instances of weak selectors in order to produce more stable results [41] and better performance than individual instances or methods [35]. It is particularly useful when the sample-to-dimensionality ratio is unfavourable, as in genomic-data samples.

Such different instances can be obtained by means of data splitting (homogeneous approach), as well as, using different selectors (heterogeneous approach). Data splitting is usually carried out by resampling the observations in multiple partitions (bagging or boosting) [11,36,42], but also through feature subspace partitioning or grouping (dimensionality reduction) [15,37,43], and simultaneously [44].

The heterogeneous approach combining results from different selectors can entail benefits from diversity. Moreover, the homogeneous approach combining results from multiple splits contributes to stability of the overall results [45].

A selector uses an inducer algorithm (classifier or regressor) to select the subset of variables that best predicts the targets of a given data split. Suboptimal solutions are provided because an approximated search in the feature space is imposed due to the computational unfeasibility of the exhaustive search. The pseudo-code of SEQENS is given in Algorithm 1.

Algorithm 1 SEQENS pseudocode

partitions ← generate *n_sequentials* random splits where *training_size%* of observations goes for train and the rest for test
for *p* in *partitions* **do**
  best_subset, *score* ← looks for the best predictive subset of variables using Sequential_Feature_Search with *max_interactions* forwards steps followed by *max_interactions* backwards steps
end for
Count the number of occurrences of each variable in all *best_subset*, disregarding low-scored selections (threshold on *score*)
return ranking of variables in descending order of the number of occurrences

2.3.1. Data splitting

Different data splitting strategies can be used by ensembles depending on the characteristics of the task to be solved and the available data. When the number of observations is low, it is advisable to use as many of them as possible to achieve the best performance and stability. In practice, this is the usual case when selecting features from genetic data because of the very high number of input variables, i.e., tens of thousands.

A resampling strategy is proposed here by splitting the whole input dataset without replacement, where each split corresponds to a

[3] scikit-learn implementation https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

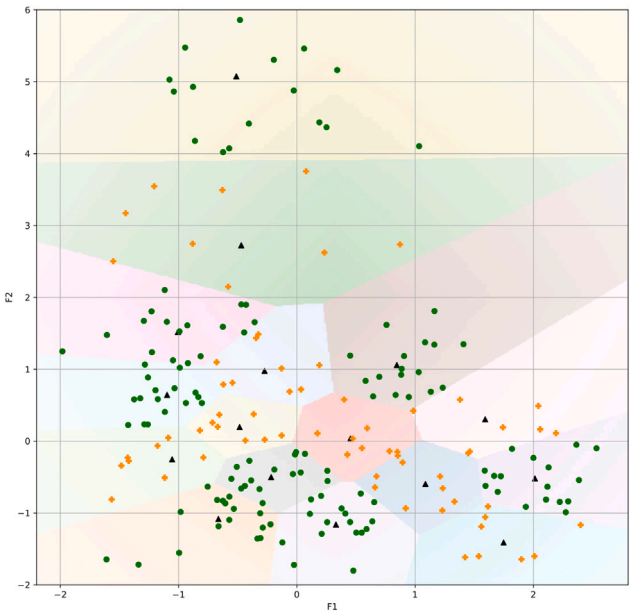

Fig. 4. 16-Clusters scenario. A 2D-example using F1 and F2 variables from 200 observations (patients). 8 clusters contain controls (green circles) and 8 contain cases (orange plus). Each background colour corresponds to a cluster region. Black triangles are the clusters centroids.

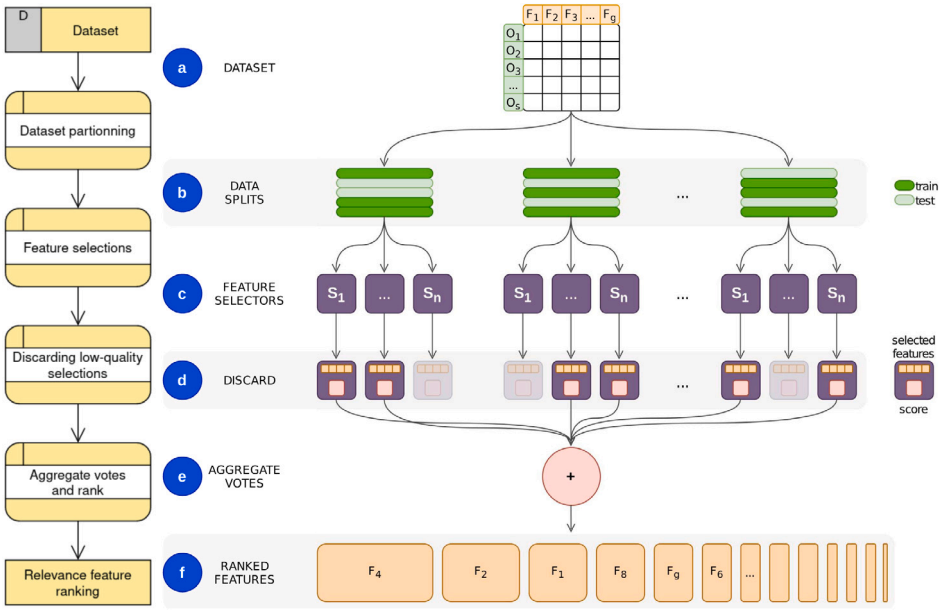

Fig. 5. (left) SEQENS Data Flow Diagram. (right) Detailed SEQENS components overview. (a) Dataset with observations (individuals) in rows and features (variables) in columns. (b) Observations are split into train and test. (c) Each split is sent to one or several feature selectors for SFS. Each selector outputs a scored subset of selected variables. (d) Discard low scored subsets. (e) Result aggregation from the selected subsets. (f) The output is a list of features ranked by relevance.

different training-test set partition. Each split will be input to a SFS algorithm to produce a subset of selected features. This is one of the key advantages of the ensemble methods: multiple training data are taken to the selectors, which allows learning from different data distributions, which in turn leads to a bias reduction [35].

The percentage of observations assigned to training and test can be adjusted to optimise performance. Smaller training sets lower the quality of the estimations while smaller test sets lower variability. A parametrisation for a hold-out error estimation is proposed in [46].

2.3.2. Sequential feature search

In real problem, it is likely that numerous variables are interacting in a complex way (e.g. through pathways) suggesting that it is fundamental to consider their relationships rather than take them individually.

The core of SEQENS is a Sequential Feature Search (SFS) greedy wrapper algorithm that implements a fast and approximate search of the most predictive combination of variables [19,20,47]. A greedy approach makes the use of wrapper methods in high dimensional spaces computationally affordable, and this is a typical situation when dealing with genetic data. Wrapper methods make no assumptions on variable distributions or on their interaction types (independence, linearity or other kernel).

A SFS forward step adds the variable that maximises the predictive power of the subset obtained in the previous step (initialised with empty subset). A SFS backward step consists in removing from the

previous subset the variable that contributes the least to the prediction (initialised with the complete set of available variables).

In this work, the proposed SFS search strategy is based on the *plus l take-away r* method presented in [47] and applies L forward steps followed by the same number of backward steps ($L = R$). Backward steps allow us to remove non-contributing variables selected in the forward stage, and the subset with the better prediction score is selected. The number of variables of the final subset will be between l and L. This strategy can be seen as a simple version of the Oscillating Search [48]. The proposed parametrisation does not explore the feature space as exhaustively as Floating or Oscillating Search, but it is faster to compute.

Setting the number of forward steps L will predetermine the maximum number of interacting variables to be explored by an SFS run. It is worth nothing that this does not imply a limit on the number of relevant variables or in the total amount of interactions that can be identified by the ensemble. Indeed, different groups of variables can be selected in different SFS runs and will be counted during the aggregation phase.

An SFS run is independently applied to each dataset split. Any estimator can be used in SFS. For each split, the training samples are used to fit the estimator and the test samples to evaluate its predictive power. The estimation score is calculated based on an objective function, i.e., accuracy for classification or the negative mean squared error (negative MSE) for regression. The objective function guides the algorithm towards the best predictive variable subset. In this paper, an SFS is also called *sequential*.

2.3.3. Aggregation criteria

An aggregation of the results coming from the multiple selections must be carried out to obtain a relevance (or importance) index for each variable, from which a ranked list can be obtained. It can include direct vote aggregation, discarding low quality selections, weighting variables or observations as a function of the selection score, or the aggregation of variable importance indexes provided by the selector used. This can improve the convenience of setting thresholds, which can be as simple as based on fixed percentages, or more novel automatic methods such as those based on data complexity measures [45]. Specific work on aggregation methods in genomic applications can be found in [49,50].

Thus, given a number of subsets of variables selected by sequentials, each of them associated with a score provided by the selector, the options considered in this work are:

1st. Discarding low quality selections. Depending on the inherent difficulty of the problem, the input sample, the objective function, as well as, other issues, selectors could not always reach a sufficient prediction quality. Thresholding can be applied on the selection score or on a percentage of the best scored subsets.

2nd. Vote aggregation. A selected variable is considered as an elector emitting a vote. The basic option is counting the number of votes received by each variable through the selected subsets (*one-feature-one-vote*). More sophisticated methods are weighting a vote by the selection score, or by the relevance position of the variable within the selected subset.

This aggregation step changes the behaviour of the whole algorithm. Since it proceeds by accumulating evidence from different selectors without taking into account possible correlations among the variables, the final set of votes can include similar or highly correlated variables that would not appear in a single selector. Each selector avoids correlated or similar variables, even if they are predictive, because together they do not contribute more than each of them individually to classification performance. Different selectors can retain different variables and therefore the final result can include all of them. That is the reason of the difference we make between selection and identification. We want to identify all the genes that influence a target condition, not to build the smallest predictive set.

2.4. Other feature selection or identification methods

The other methods with which SEQENS is compared are briefly described in this section. In all these techniques, the variables can be ranked by relevance according to their respective computed feature-importance statistics.

Analysis of variance test (ANOVA), Pearson correlation (Pearson), and statistical Welch's two samples t-test (Welch) are model-free univariate techniques that do not take into account variable interactions (univariate). ANOVA is a test based on analysis of variance. The Pearson correlation is computed to measure the linear dependence between each variable taken separately and the target. Welch's t-test test checks if two samples of unequal variances have the same mean to separate cases and controls using this difference in mean.

Minimum Redundancy Maximum Relevance (mRMR) feature selection [51,52] is a model-free approach looking for variables that maximise their relevance to the target (correlation) but minimise their redundancy with other variables using mutual information criterion. It is an iterative process in which, for a given iteration, the algorithm selects the variable presenting both the best relevance with the target and the minimum redundancy with the variables already selected in the previous stages.

ReliefF approaches [53–56] are also model-free. They are based on the idea that in a region of the hyperspace where the observations of a given class are in the majority, the nearest neighbours of the same class will be closer than the neighbours of the other classes. For a given observation of a given class, the algorithm calculates the difference between the distance from the observation to its nearest neighbour of the same class subtracted to the distance between the observation and the nearest neighbour of the other classes. In this process, several neighbours can be used to improve the stability of the method. The feature importance is then computed accumulating the absolute differences in each dimension (variable). Relevant variables are likely to accumulate higher differences than noise variables. In this paper, the MultiSURF implementation is tested.

Lasso linear modelling has been extensively used in bioinformatics in the task of feature selection [57–61]. Lasso considers that the variables interact in a linear way (weighted sum) and calculates the coefficients (weight) of each variable using a least square minimisation enriched with a constraint on the sum of the coefficients. The weight associated to each variable is its feature importance. Adjusting the constraint allows the algorithm to select more or less relevant features, setting the weight of the irrelevant variables to 0. The higher the constraint, the fewer the number of variables selected. On the contrary, a null constraint is equivalent to apply directly a linear regression and to make no selection.

SVM-based recursive feature elimination (SVM-RFE) is a combination of a linear Support Vector Machine inducer with a recursive feature elimination process where, at each iteration, a percentage of the less important features are removed [62–64]. In this iterative process, a linear SVM is trained and tested following a cross-validation scheme at each iteration. Variables are sorted according to their absolute linear weight. A percentage of the less weighted (less important) are discarded and the process is repeated.

Like SEQENS, Random Forest (RF) [28] and Gradient Boosting (GBoost) [29] are ensemble methods. They both combine multiple decision trees. RF computes the trees from splits of the dataset following a bagging strategy (like SEQENS). GBoost builds a sequence of decision trees (boosting) where the errors made on one tree are used to weight the observation importance for the next tree. Both methods aggregate individual tree feature selections into a collective decision to rank features by importance.

Table 1 lists all the methods tested, provides some keywords to identify which category of approximations they belong to and specifies the inducer (or not) they depend on.

Table 1
Description of the feature identification algorithms tested.

Algorithm	FI type	Inducer
ANOVA	Filter, univariate	Model free
Pearson	Filter, univariate	Model free
Welch	Filter, univariate	Model free
mRMR	Filter	Model free
MultiSURF	Filter	Model free
Lasso	Embedded	Constrained linear model
SVM-RFE	Embedded	Linear support vector machine
RF	Wrapper, ensemble (bagging)	Decision tree
GBoost	Wrapper, ensemble (boosting)	Decision tree
SEQENS	Wrapper, ensemble (bagging)	k-neighbours, sequential search

2.5. Cliff, a ranking evaluation measure

Measuring the quality of a feature identification ranking allows us to directly and accurately assess the performance of a given technique, as well as compare different techniques. A metric named *Cliff* is presented here. It aims to quantify the quality of a feature identification list ranked by relevance when a groundtruth of relevant variables is available.

Cliff is designed to score 1 when all the relevant variables (groundtruth) are located in the first positions of the ranking. The score tends towards 0 as the relevant variables move away from the first positions.

Let $\mathcal{L}$ be a list of variables ranked by relevance merits, and let $\mathcal{L}'$ be the set of positions of the relevant variables in $\mathcal{L}$. The cliff score of $\mathcal{L}$ is the sum of position scores s of the relevant variables in $\mathcal{L}$, as expressed in Eqs. (2) and (3), where R is the number of relevant variables. The function s assigns a score to a position p on the list.

$$\text{score}(\mathcal{L}) = \sum_{p \in \mathcal{L}'} s(\mathrm{R}, p) \tag{2}$$

$$s(\mathrm{R}, p) = \begin{cases} \frac{1}{\mathrm{R}} - \frac{\alpha}{2}(2p - \mathrm{R} - 1), & p \leq \mathrm{R} \\ s(\mathrm{R}, \mathrm{R})(p - \mathrm{R} + 1)^{-\beta}, & p > \mathrm{R} \end{cases} \tag{3}$$

We considered variables in the top R positions to have a different treatment from the rest. Hence, s is a piecewise function made up of two decreasing sub-functions whose slopes can be independently tuned using two coefficients denoted by α and β. The first part ($p \leq \mathrm{R}$) applies to relevant variables found within the top R positions, and decreases linearly or keeps constant. Thus, a variable retrieved in position $p \leq \mathrm{R}$ will be assigned an higher or equal score than a variable in position $p + 1$. The second part ($p > \mathrm{R}$) decreases following a negative power function with exponent $-\beta$ and will apply to variables retrieved over position R.

s has the shape of a cliff, as shown in the example of Fig. 6, where R is set to 10. By setting $\alpha = 0$ and $\beta \to +\infty$ a Heaviside function is obtained (green line). This configuration could be used to obtain a cliff score equal to the percentage of relevant variables retrieved in the first R positions, regardless of the relevant variables positioned clearly over R. In this case, two algorithms with the same number of relevant variables found within the top R positions will obtain the same score. Nevertheless, if one wish all the relevant variables to contribute to the final score, a configuration with $\alpha > 0$ and β being a positive real value must be considered, as depicted by blue and black lines. $\alpha = 0.008$ and $\beta = 0.5$ are the settings chosen in the experiments of this paper (black line).

The higher the α, the higher the score mass is moved from the second half to the first half of the R-top positions amplifying the importance of the very first positions. The higher the β, the faster the decrease in the score after the R position. α and β must be positive real values, and α has to be constrained to the range $[0, \frac{2}{\mathrm{R}(\mathrm{R}-1)}]$ to avoid s crossing the abscissa axis.

Variables not being selected (or receiving no votes in a voting scheme) by the FI algorithm are considered at the ∞ position and are assigned $s = 0$. In case of ties of relevance merits, the average position of the tied variables is assigned to all of them.

3. Results

For the experiments, three sets of case/control targets (groundtruths), named hyperplane, hypersphere and 16-Clusters, have been generated from interactions of 10 relevant variables (generative variables) following the three different interaction patterns proposed in Section 2.2. This process has been carried out using all the 27,887 observations of the E-MTAB-3732 database.

The goal of the experiments is to measure the ability of several state-of-the-art algorithms, including SEQENS, to identify and locate the relevant variables in the top positions of their ranks in the three proposed scenarios. The cliff metric presented in Section 2.5 is used as a quality indicator of the ranks obtained.

For a given scenario, several sample-to-dimensionality ratios (SDR) are explored. A different number of observations, ranging from 300 to 3000, are combined with four amounts of variables, 854, 3417, 13,669 and 54,675 (total amount of variables in the database), which corresponds to SDR ranging from 0.0055 to 3.5.

To select the subsets of observations, random subsampling of the 27,887 observations available (already labelled) is carried out. Each set of variables to be tested include the 10 relevant variables plus the corresponding number of noise variables randomly selected. To provide information about the stability, each presented result is achieved from ten runs, where each run is performed on a different subsampling. The resulting ten scores are averaged, and the mean and 95% confidence interval (CI) are provided.

The following two sections detail the configurations of the FI methods evaluated, and then, Section 3.3 presents a comparative analysis of their performances and stability.

3.1. SEQENS configuration

The SFS greedy algorithm was set to perform 10 forward steps (parameter *max_interactions* in Table 2), reflecting the quantity of relevant variables that are actually interacting. In a real task, such a number would not be known, therefore, this parameter should be set based on a trade-off between a maximum number of underlying interactions expected and the computational cost associated to the search in high dimensionality spaces. As discussed in Section 2.3.2, the number of backward steps was set equal to the forward steps. If a draw occurs among candidate variables at any stage, one of the variables is randomly chosen.

A hundred independent runs of the SFS algorithm, or *sequentials*, was considered to compose one ensemble. Each sequential is run on a different sample split. The higher the number of sequentials, the higher the number of votes to be spread out, which would lead to the overall result to converge. Ideally, the votes received by the noise variables should be uniformly distributed, while the relevant variables should accumulate votes, avoiding the noise variables to reach the first positions of the ranking. This should be the case where the noise variables do not have a significant influence on the target.

Within a given split, 20% of the observations was used for training, and the remaining 80% for test (hold out estimation) preserving the case/control ratio within both sets. This parameter was optimised using a single subsampling of 1200 observations and 54,675 variables. This subsampling was not used to compute the results.

A single inducer was used, a *k*-Nearest Neighbours regressor weighted by inverse distance [65] with $k = 5$ using the coefficient of determination R^2 as objective function (score). The Eq. (4) presents R^2 score. It is based on the ratio between the error made by the predictor and the error that a simple predictor always returning the mean of the target values would obtain. y_i is the groundtruth target value of the i-est sample, $\hat{y}_i$ is the estimation made by the predictor and $\bar{y}$ is the mean of

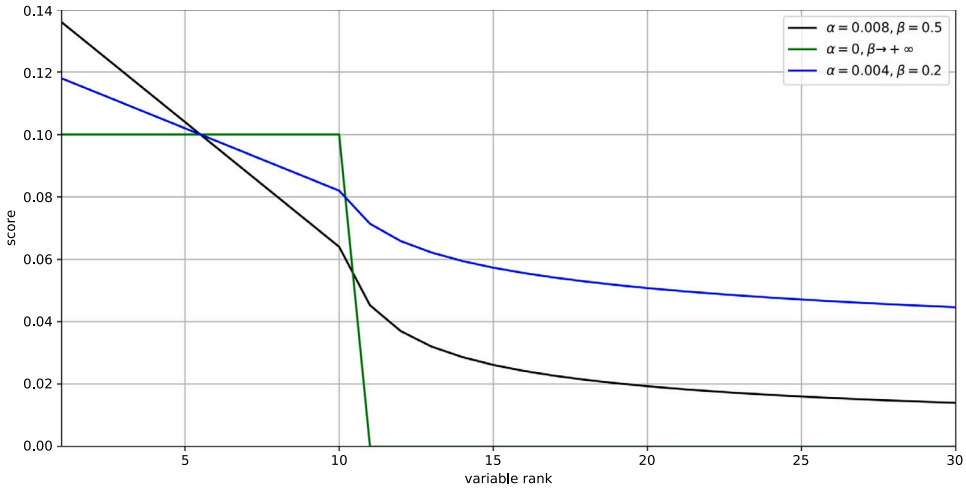

Fig. 6. Example of three configurations of the function s: (green line) A Heaviside function that assigns all the score mass to the top R positions equally distributed; (black line) configuration used in our experiments; and (blue line) intermediate configuration. Compared to the black curve, the two last configurations reward relevant variables found over R in different magnitudes and the score assigned to any position p is higher than the score of position $p+1$.

the target values. A R^2 of 1 is equivalent to a perfect estimation while a R^2 inferior or equal to 0 indicates that the predictor is making more error than the predictor returning the mean of the target values.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{4}$$

The selector is embedded in the SFS algorithm. Thus, as a result of a SFS execution, the subset of features exhibiting the maximum R^2 is selected.

Regarding the aggregation criteria, on one hand, *one-feature-one-vote* was selected due to its simplicity and interpretability, thereupon no weighting function was applied. On the other hand, thresholding on the score of the selected subsets provided by the sequentials was applied to reject the lower quality ones, which can be seen as a simple filter to improve results. In all the experiments, 20% of the best scored sequentials are kept for polling (and the rest are rejected).

3.2. Other algorithm configurations

Table 2 summarises the implementations and configurations of the methods included in the experiments. Some of them have no parameters, such as ANOVA, Pearson, and Welch t-test. Moreover, the MultiSURF implementation tested, representing ReliefF approaches, has no parameters to set along with the mRMR implementation based on mutual information quotient (MIQ) schemes. For Lasso, SVM-RFE, GB, and RF, a few parameters were optimised. The subsamplings used for optimisation were not used for performance evaluation.

The main Lasso parameter, α, is the weight associated to the constraint on the sum of the linear coefficients. The optimal value was calculated by exploring within the range 10^{-6} to 10^{1} the α value that maximises the cliff measurement. In order to do this, tests on four subsamplings per scenario (twelve samples) of 1500 observations each, have been run. The optimal α was computed as the average of all the twelve runs, and as a result, it was set to 0.008.

SVM-RFE relies on two main parameters. Firstly, *step*, is the number of variables discarded at each step of the recursive feature elimination. This was set to 0.05, meaning that 5% of the less important variables are discarded in each iteration. Secondly, parameter C, is the SVM margin. Its optimisation was carried out in the same conditions as the one from Lasso, and, as a result, its value was set to 0.002. The support vector machine used in the experiments had a linear kernel.

Given that both Gradient Boosting (GBoost) and Random Forest (RF) are ensemble methods, as with SEQENS, the number of trees was set equal to the number of sequentials in SEQENS, i.e., 100 estimators. Each tree was built with all the available variables. In RF, a bootstrap of 20% of the samples was used to train the trees (parameter *max_samples* in Table 2), as in SEQENS, and no limit was applied on the tree depth. In GBoost, all the observations were used to fit the trees and the maximal tree depth was 3 (default values in the implementation used). RF and GBoost are both used in their regression version to be consistent with SEQENS (using a k-nearest neighbours regressor).

3.3. Performances evaluation

Each row in Fig. 7 shows the results obtained for the four variable amounts tested (854, 3417, 13,669, and 54,675) on the three generated scenarios (hyperplane, hypersphere, and 16-Clusters). Each group of variables includes the variables of the smaller group plus new variables randomly selected. All the four groups include the same 10 relevant variables. The sample size is indicated in the horizontal axis and the obtained cliff-score in the vertical axis. Each method is represented by a different colour line. Each point of a curve is the average cliff-score obtained from ten experiments of its corresponding method coming from ten different subsamplings of the dataset. The 95% confidence interval is indicated in low-tone background colour around the solid mean line. Any given subsampling includes the observations of its respective smaller subsampling. The black line represents the results of a random identification, where each point was calculated following the same subsampling and averaging procedure as in other methods.

When comparing performance in different scenarios, different rankings of methods are observed. Globally, SVM-RFE, Lasso, RF, GBoost and SEQENS are the ones that provide better results.

ANOVA and Welch do not provide more information than a random identification in any scenario, and, in general, Pearson shows low scores except for small amounts of variables, where its results are still poor compared to the rest of the methods. Thus, as expected, the high dimensionality is revealed to be a major obstacle for pure filter univariate approaches.

MultiSURF and mRMR provide scores superior to Random and Pearson methods but remain significantly lower than the best algorithms. Besides, they do not clearly benefit from the increase in the number of individuals.

In the hyperplane scenario (left column), Lasso presents the best performance, followed by SEQENS. Lasso is able to identify all the relevant variables in top positions of the rank (cliff score = 1) with a sufficient number of observations, except when dealing with 54,675 variables. SEQENS fails to discover two out of the ten variables (always the same two), independently of the amount of noise variables it deals

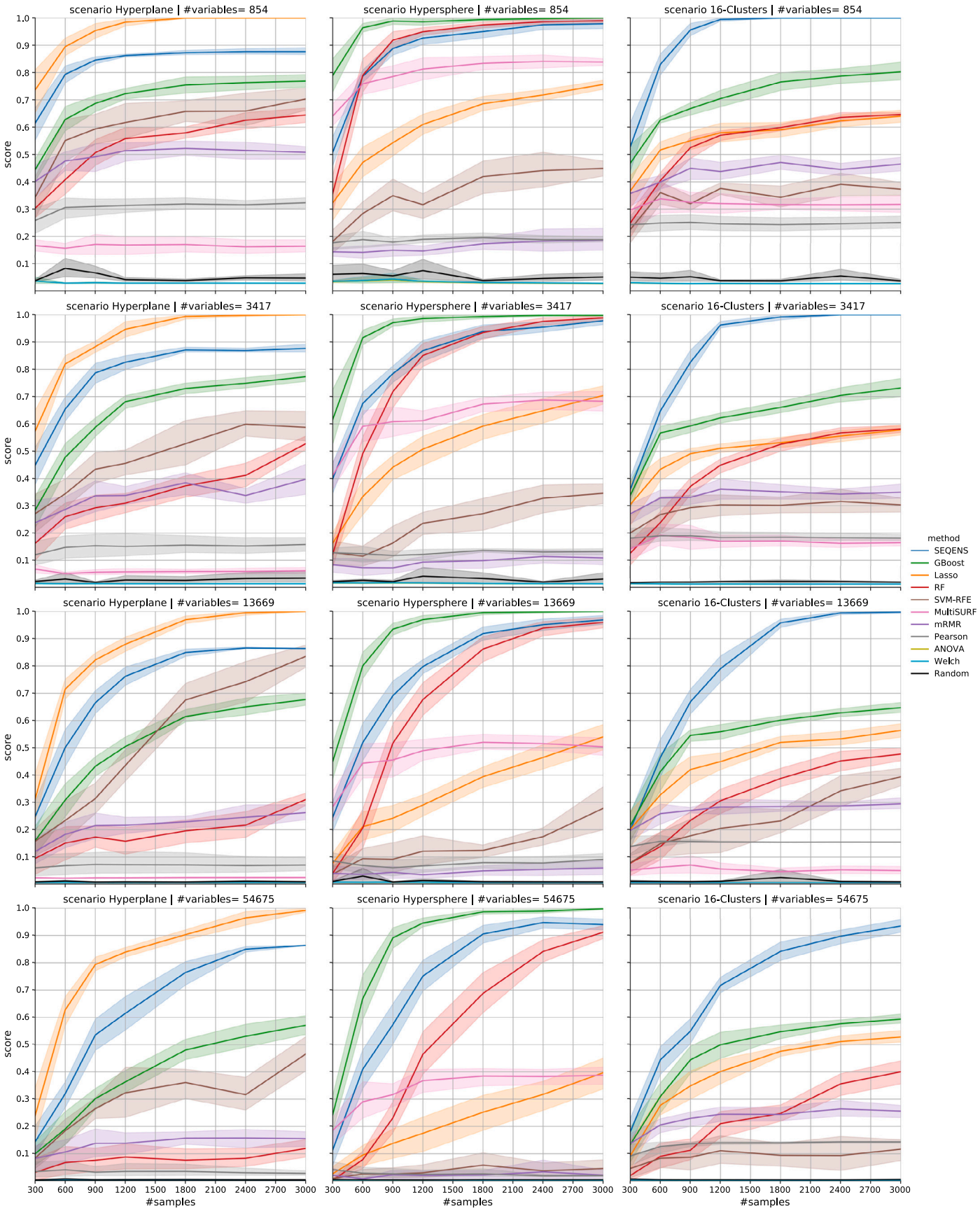

Fig. 7. Cliff-scores obtained by each FI method tested for different SDR on the three interaction scenarios. The columns are the scenarios, the rows are the number of variables used, and the sample size is indicated in the horizontal axis of the plots. The mean of 10 runs on different subsamplings and the 95%-CI (background area) are depicted. For computation time reasons (see Table 4), MultiSURF was not calculated in the Hyperplane and 16-Clusters scenarios because the expected results would be worse than with 13,669 variables where they are already close to a random identification.

Table 2
Algorithm implementations and configurations.

Algorithm	Implementation	Configuration
ANOVA	*sklearn.feature_selection.f_classif*[a]	Default
Pearson	*scipy.stats.pearsonr*[b]	Default
Welch	*scipy.stats.ttest_ind*[b]	Default
mRMR	*mrmr.mrmr_classif* https://github.com/smazzanti/mrmr	Default
MultiSURF	*skrebate.MultiSURF* https://epistasislab.github.io/scikit-rebate	Default
Lasso	*sklearn.linear_model.Lasso*[a]	$\alpha = 0.008$[c]
SVM-RFE	*sklearn.svm.LinearSVR, sklearn.feature_selection.RFE*[a]	$C = 0.002$[c] $step = 0.05$
Random Forest	*sklearn.ensemble.RandomForestRegressor*[a]	$n\_estimators = 100$ $max\_samples = 0.2$
Gradient Boost	*sklearn.ensemble.GradientBoostingRegressor*[a]	$n\_estimators = 100$
SEQENS	*seqens.Seqens* https://www.kaggle.com/itiresearch/seqens	$n\_sequentials = 100$ $max\_interactions = 10$ $training\_size = 0.2$[c]

[a] *scikit-learn* library [66] version 0.24.2 (https://scikit-learn.org/stable/user_guide.html).
[b] *scipy* library [67] version 1.6.3.
[c] These parameters have been optimised.

with. In this scenario, SVM-RFE obtains competitive results, particularly with 13,669 variables. However, as denoted by its CIs, it is the least stable algorithm, a phenomenon which was also observed during the optimisation of its margin C parameter (Section 3.2).

In the hypersphere scenario (centre column), ensemble methods (GBoost, RF and SEQENS) show the best scores. With 54,675 variables, GBoost reaches a 0.9 cliff-score using only 900 observations while SEQENS, the second ranked method, requires 1800 observations. It should be noted that MultiSURF performs much better in this scenario than in the other scenarios.

In the 16-clusters scenario (right column), where variable interactions are more complex, SEQENS clearly outperforms the rest of the methods, and it reaches full identification (10 relevant variables on top 10 positions) in some configurations. Only with the smallest sample sizes (300 and 600 observations), do the SEQENS, Lasso and GBoost confidence intervals overlap.

In the most unfavourable SDR (e.g., 300 observations for 54,675 variables), the scores obtained are low even for the best algorithms (GBoost, Lasso, SEQENS). Only one or two variables are expected to be found in the first ten positions of the ranking. Nevertheless, in all the experiments, the scores show a significant improvement when stepping from 300 to 600 observations. The magnitude of this improvement slows down as the sample size increases until the plateau of the curves is reached. This reinforces the importance of collecting a sufficient number of observations, and also shows that an upper boundary on observations exists for each method. As can be noted, this boundary depends not only on the number of variables found in the database (SDR) but also on the type of scenario, i.e., the type of interactions among the relevant variables.

In practice, the interactions between relevant variables (scenario) are not known a priori, either in number or type. Therefore, it is not possible to select the most suitable method according to this criterion. Thus, considering the overall performances on the three different synthetic diseases tested, SEQENS turns out to have a higher generalisation power and can be considered the most appropriated method to be used.

To argue this point, the average performance of the algorithms on the three fictitious diseases (hyperplane, hypersphere, and 16-Clusters) is illustrated in Fig. 8. Each point of the curves represents the cliff-score mean of all the 30 subsamplings (ten per scenario), and its corresponding confidence interval is depicted. In general, up to 900 observations, SEQENS is sharing with GBoost the best average score. Over 900 observations, SEQENS obtains the best average score on the four datasets and its distance with the second best method increases as the dimensionality increases.

Stability is an important characteristic of feature identification. Ideally, feature rankings should remain the same in the presence of slight perturbations of the dataset. In this work, the stability is measured as the variation of the cliff score among different subsampling of a population. On average over the three scenarios considered, SEQENS exhibits a smaller CI than the other methods, which indicates that it produces rankings of relevant features that are more stable and less sensitive to data variations, and therefore, it can be considered as the more stable gene identification option. To illustrate this point, Table 3 shows the average cliff scores across all scenarios with 54,675 variables. The 95% confidence interval is depicted for the four methods with the highest mean score at 3000 observations.

3.4. Computational considerations

This section aims at giving an idea of the complexity of algorithms in a concrete example. It is difficult to be completely fair given the variety of implementations, and that some incorporating elements of parallelism and optimisations into the code while others do not. The computing time for each method is presented using the dataset with 1200 observations and 54,675 variables.

The computer used is a 16 core Intel Xeon Processor (Skylake, IBRS) 2.3 GHz with 32 GB RAM. Table 4 presents the average elapsed time for the 10 runs separating results per disease.

SEQENS clearly has the heaviest computational load. Despite this, in practice, identifying the relevant variables is not a calculation that needs to be repeated many times. It therefore seems worth it to allow several hours or days of calculation in order to obtain a better quality gene identification.

Moreover, it is worth mentioning that SEQENS is a perfectly parallel algorithm, which means that the computation time decreases in an almost proportional way with the quantity of available cores.

Each SFS performed by SEQENS will train and test $max\_interactions \times F$ models, F is the number of variables and $max\_interactions$ the number of forwards steps. The quantity of models to train and test increases linearly with the number of variables and the maximum interaction size desired. Complete SEQENS algorithmic complexity is then determined by the inducer embedded into the sequential feature search and how it behaves when the number of observations or variables increases.

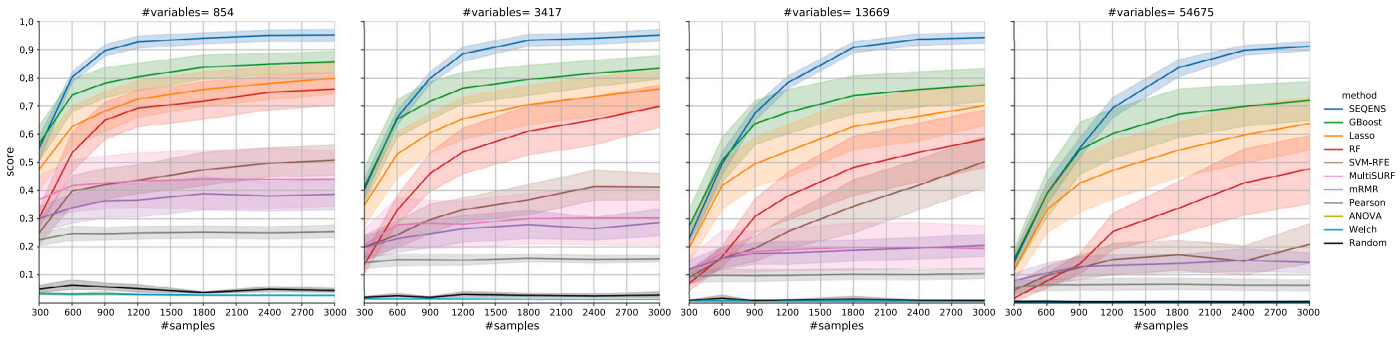

Fig. 8. Averaged cliff-scores of the three scenarios. The columns are the number of variables in the dataset. The mean and 95%-CI from all the 30 runs are shown. SEQENS obtains the best average score over 900 observations and exhibits a smaller CI than the other methods.

Table 3
Mean cliff scores across all scenarios with 54,675 variables. 95% CI interval is shown for SEQENS, GBoost, Lasso and RF methods. Methods are sorted by decreasing mean score for 3000 individuals. For each sample size (column), the method with the highest mean score is indicated in bold.

#samples method	300	600	900	1200	1800	2400	3000
SEQENS	0.146 ±0.033	**0.391** ±0.038	**0.552** ±0.035	**0.694** ±0.036	**0.837** ±0.030	**0.898** ±0.018	**0.913** ±0.016
GBoost	**0.159** ±0.041	0.390 ±0.082	0.546 ±0.093	0.603 ±0.092	0.671 ±0.082	0.699 ±0.075	0.721 ±0.071
Lasso	0.120 ±0.049	0.333 ±0.083	0.427 ±0.101	0.472 ±0.102	0.543 ±0.099	0.598 ±0.099	0.639 ±0.094
RF	0.019 ±0.018	0.079 ±0.025	0.140 ±0.036	0.254 ±0.066	0.337 ±0.097	0.427 ±0.114	0.477 ±0.119
SVM-RFE	0.047	0.098	0.126	0.155	0.171	0.149	0.209
mRMR	0.080	0.107	0.130	0.135	0.141	0.152	0.145
Pearson	0.056	0.066	0.065	0.067	0.067	0.064	0.064
Random	0.006	0.006	0.005	0.005	0.006	0.005	0.006
Welch	0.004	0.004	0.004	0.004	0.003	0.003	0.003
ANOVA	0.004	0.004	0.004	0.004	0.003	0.003	0.003

Table 4
Algorithms computing mean time in seconds for the 10 runs with 1200 observations and 54,675 variables.

Algorithm	Hyperplane	Hypersphere	16-clusters
ANOVA	<1	<1	<1
Pearson	4	4	4
Welch	10	10	10
Lasso	29	43	34
RF	293	439	409
SVM-RFE	459	458	460
mRMR	1496	1500	1493
GBoost	1677	1679	1671
MultiSURF	2366[a]	9256	2160[a]
SEQENS	50,972	46,910	37,173

[a]Obtained with 13,669 variables so the duration is underestimated.

4. Discussion

To a greater or lesser extent, the performances of all methods decrease as the dimensionality (amount of variables) increases. In addition, performance tend to increase as the sample size increases, with the exception of MultiSURF and mRMR methods. FI methods reach an asymptotic plateau more or less quickly as the sample size increases. These observations are coherent with the well-known curse of dimensionality that affects the induction algorithms.

Depending on the scenario observed, the optimal method is different, which is a strong signal in favour of the use of ensemble methods which take advantage of the diversity of points of view. With this perspective, this work presents the highly parallelisable ensemble machine learning algorithm SEQENS for feature identification in high dimensionality spaces, such as genomic data. On microarray data, SEQENS shows good performance in the three proposed scenarios and, on average, it worked better and is more stable than the methods it is compared with.

Special attention has been paid to comparing algorithms as fairly as possible. We are aware that variations and improvements of the methods tested can be found in the literature. Meanwhile, SEQENS remains wide open to optimisations in many different ways. In this work, only one parameter of SEQENS has been optimised.

Increasing diversity should lead to significant improvements of SEQENS. These could be achieved by means of:

- Combining inducers. The algorithm presented in this paper uses a single inductor (k-nearest neighbours) but it is possible to combine several of them. Some tentative experiments suggested it would strengthen gene identification in different scenarios.
- Exploring new selector types. Current implementation of SEQENS is linked with sequential feature search, other selectors like genetic or swarm approaches could be tested and even combined.
- Extending data splitting. Combining results from multiple sample splits is the base for stability in ensembles. Variable space partitioning could also contribute to improving the results.

Regarding the SEQENS parametrisation, while additional parameters can be studied (such as those related to the selector parametrisation or result aggregation), three parameters have been tackled in this work:

- The number of sequentials (*n_sequentials*) to run will determine the degree of convergence of the results. It is linearly related to the computing time. Despite the fact that a trade-off between runtime and convergence should be assumed, its setting is not critical. One hundred has been a satisfactory amount in our experiments.

- The training size (*training_size*) is the percentage of samples used to train an inducer, the remaining samples going to its evaluation. In this work, it has been optimised, and only 20% was set aside for training. It is consistent with the fact that the ensemble method uses weak selectors, where preference is given to test rather than train.
- The maximum number of interactions (*max_interactions*) parameter corresponds to the number of forward steps in the Sequential Feature Search algorithm. It can be interpreted as the maximum number of interactions within a group of variables that can be detected. Small values will provide only a partial view of the entire interaction while large values increase the computational cost and tend to introduce extra noise variables into the selection (despite this, it does not necessarily affect the results). To configure the experiments in a proper manner, its value is set to ten, according to the number of relevant features defined in the groundtruth.

The Lasso algorithm offers the best performance in the hyperplane scenario. This is as expected because it is intrinsically designed to detect linearities (minimisation of a quadratic error with respect to a linear function). Nevertheless, this result is conditioned by the optimal setting of its main parameter α, which allows us to adjust the quantity of variables to be considered as relevant. With the optimisation methodology of α followed in this paper, the scores presented for Lasso were, on average, the best possible. Nevertheless, in practice, for a given dataset with no groundtruth, it is not possible to calculate the optimal value of α directly, but only estimate it [68] (e.g., maximising the classification score in a cross-validation strategy). The same reasoning applies to SVM-RFE method but, in this case, its performance is significantly lower.

It is worth highlighting that GBoost, which is an ensemble approach like SEQENS, achieves the second best performance on average. This is one more signal confirming the strength of ensemble methods for gene identification. With Random Forest, the overall performance is lower but it is however important to observe that it works well in Hypersphere scenario.

In our benchmark, FI using classic univariate statistical methods do not work in the presence of a several hundred variables or more, even when the number of observations is large [8]. In almost all the cases, FI using statistical tests is not distinguishable from a random selection.

Regarding the synthetic disease generation (groundtruth), the mechanism of cause–effect underlying a real disease and the variables involved, and their interactions, is very difficult to fully understand, and hence, to translate into a synthetic scenario. Nevertheless, the three proposed scenarios cope with three classic tasks in machine learning, emulating a wide range of tasks of the real world. In fact, the results presented clearly show that different methods perform different in different scenarios. In this sense, SEQENS performs better than the rest as it seems to be a more generalisable method. More work on new scenarios, particularly those based on real disease knowledge, when available, could be done.

The presented Cliff evaluation measurement takes advantage of having synthetic disease scenarios, which can help with parameter optimisation. Cliff measures the quality of the variable identification directly. For various reasons, this can be better than using prediction tests if a groundtruth is available. On one hand, prediction involves additional tasks such as removing redundant features and training models. On other hand, many times, the immediate goal of FI is to discover biomarkers associated to a phenotype, which involves post-selection biological analysis and cohort studies. This does not necessarily entail a prediction task as a target. In this sense, the Cliff measurement can be easily adapted to tolerate a wide range of top positions of the rank as relevant.

Although the computational load of SEQENS is costly in absolute terms, and significantly higher than the rest of the methods, it is perfectly affordable with currently available resources. In practice, e.g., in biomarker discovery, the process of finding relevant genes, SNP, metabolites, or whatever kind of attribute, should be run only once (or a small number of times) before sending the results to a laboratory for validation.

5. Conclusions

In this article, SEQENS, a method for relevant gene identification is presented and compared with other state-of-the-art approaches.

In order to directly measure the ability of algorithms to identify relevant variables, a groundtruth was generated as three fictitious diseases. In practice, it is difficult to have a reliable groundtruth because the relevant variables are precisely the object sought. The disease generation methodology presented is a first proposal, and further fictitious diseases could be generated according to more realistic biological knowledge.

None of the methods presented outperforms the others in all the scenarios (diseases). Therefore, on average, SEQENS identifies significantly better the genes associated to the three proposed fictitious diseases when a minimum number of individuals are available. In other words, it generalises better to different diseases (scenarios) than the other methods it is compared with. It also gives better stability. As in practice it is difficult to know beforehand the kind of interaction between relevant variables, this is a strong argument towards the use of SEQENS in gene identification.

It is essential to continue collecting data from patients in order to build up large databases of several thousand observations. As the results of this paper suggest, the more observations, the better the gene identification results.

An implementation is made available to the community at https://www.kaggle.com/itiresearch/seqens. It allows the user to adjust the main parameters of the algorithm.

SEQENS computational cost is high but should not be seen as a limitation. Indeed, the identification of genes is a task that should only be carried out once and is of great use for medical research since it can help focus on genes that may have an influence on a particular disease. It is also an important step towards the development of decision support tools to bring artificial intelligence into clinical practice.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by Generalitat Valenciana through IVACE (Valencian Institute of Business Competitiveness) distributed nominatively to Valencian technological innovation centres under project expedient IMAMCN/2021/1.

It was also funded by the Cervera Network of Excellence Project in Data-based Enabling Technologies (AI4ES), co-funded by the Centre for Industrial and Technological Development, E.P.E. (CDTI) and by the European Union through the NextGenerationEU Fund, within the Cervera Aids program for Technological Centres, with the expedient number CER-20211030.

References

[1] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature Selection for High-Dimensional Data, Springer, 2015, google-Books-ID: QgO0CgAAQBAJ.

[2] R. Alanni, J. Hou, H. Azzawi, Y. Xiang, A novel gene selection algorithm for cancer classification using microarray datasets, BMC Med. Genomics 12 (1) (2019) 10.

[3] A. Dabba, A. Tari, S. Meftali, R. Mokhtari, Gene selection and classification of microarray data method based on mutual information and moth flame algorithm, Expert Syst. Appl. 166 (2021) 114012, http://dx.doi.org/10.1016/j.eswa.2020.114012, URL https://www.sciencedirect.com/science/article/pii/S0957417420307855.

[4] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, M. Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, Comput. Statist. Data Anal. 143 (2020) 106839, http://dx.doi.org/10.1016/j.csda.2019.106839, URL https://www.sciencedirect.com/science/article/pii/S016794731930194X.

[5] M. Wang, S.-H. Lo, T. Zheng, I. Hu, Interaction-based feature selection and classification for high-dimensional biological data, Bioinf. (Oxford, England) 28 (2012) 2834–2842, http://dx.doi.org/10.1093/bioinformatics/bts531.

[6] B.W. Kulohoma, F. Marriage, O. Vasieva, L. Mankhambo, K. Nguyen, M.E. Molyneux, E.M. Molyneux, P.J.R. Day, E.D. Carrol, Peripheral blood RNA gene expression in children with pneumococcal meningitis: a prospective case-control study, BMJ Paediatr. Open 1 (1) (2017) e000092.

[7] K. Schramm, Gene Expression Studies (Ph.D. thesis), 2016, URL http://nbn-resolving.de/urn:nbn:de:bvb:19-207031.

[8] M. Jeanmougin, A. de Reynies, L. Marisa, C. Paccard, G. Nuel, M. Guedj, Should we abandon the t-test in the analysis of gene expression microarray data: A comparison of variance modeling strategies, PLOS ONE 5 (9) (2010) 1–9, http://dx.doi.org/10.1371/journal.pone.0012336.

[9] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28, http://dx.doi.org/10.1016/j.compeleceng.2013.11.024.

[10] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, M.K. Rasel, A review of ensemble learning based feature selection, IETE Tech. Rev. 31 (3) (2014) 190–198, http://dx.doi.org/10.1080/02564602.2014.906859.

[11] A.A.-B. Veónica Bolón-Canedo, Ensembles for feature selection: A review and future trends, Inf. Fusion 52 (2019) 1–12, http://dx.doi.org/10.1016/j.inffus.2018.11.008.

[12] H. Liu, H. Motoda, L. Yu, Feature selection with selective sampling, in: Proceedings of the Nineteenth International Conference on Machine Learning, 2002, pp. 395–402.

[13] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, Filter methods for feature selection – a comparative study, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2007, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 178–187.

[14] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1) (1997) 273–324, http://dx.doi.org/10.1016/S0004-3702(97)00043-X, URL http://www.sciencedirect.com/science/article/pii/S000437029700043X.

[15] J.C. Ang, A. Mirzal, H. Haron, H.N.A. Hamed, Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection, IEEE/ACM Trans. Comput. Biol. Bioinform. 13 (5) (2016) 971–989, http://dx.doi.org/10.1109/TCBB.2015.2478454.

[16] Z.M. Hira, D.F. Gillies, A review of feature selection and feature extraction methods applied on microarray data, Adv. Bioinf. 2015 (2015) http://dx.doi.org/10.1155/2015/198363, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4480804/.

[17] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Inform. Sci. 282 (2014) 111–135, http://dx.doi.org/10.1016/j.ins.2014.05.042, URL https://www.sciencedirect.com/science/article/pii/S0020025514006021.

[18] P. Drotár, J. Gazda, Z. Smékal, An experimental comparison of feature selection methods on two-class biomedical datasets, Comput. Biol. Med. 66 (2015) 1–10, http://dx.doi.org/10.1016/j.compbiomed.2015.08.010, URL https://www.sciencedirect.com/science/article/pii/S0010482515002917.

[19] P. Pudil, F.J. Ferri, J. Novovicova, J. Kittler, Floating search methods for feature selection with nonmonotonic criterion functions, in: Proceedings of the 12th IAPR International Conference on Pattern Recognition, 3 - Conference C: Signal Processing (Cat. No. 94CH3440-5), 2, 1994, pp. 279–283, http://dx.doi.org/10.1109/ICPR.1994.576920, vol. 2.

[20] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognit. Lett. 15 (11) (1994) 1119–1125, http://dx.doi.org/10.1016/0167-8655(94)90127-9, URL http://www.sciencedirect.com/science/article/pii/0167865594901279.

[21] M. Dashtban, M. Balafar, Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts, Genomics 109 (2) (2017) 91–107, http://dx.doi.org/10.1016/j.ygeno.2017.01.004, URL https://www.sciencedirect.com/science/article/pii/S0888754317300046.

[22] A.K. Das, S. Das, A. Ghosh, Ensemble feature selection using bi-objective genetic algorithm, Knowl.-Based Syst. 123 (2017) 116–127, http://dx.doi.org/10.1016/j.knosys.2017.02.013, URL https://www.sciencedirect.com/science/article/pii/S0950705117300801.

[23] C. Peng, X. Wu, W. Yuan, X. Zhang, Y. Zhang, Y. Li, Mgrfe: Multilayer recursive feature elimination based on an embedded genetic algorithm for cancer classification, IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (2) (2021) 621–632, http://dx.doi.org/10.1109/TCBB.2019.2921961.

[24] L. BrezoÄánik, I. Fister, V. Podgorelec, Swarm intelligence algorithms for feature selection: A review, Appl. Sci. 8 (2018) 1521, http://dx.doi.org/10.3390/app8091521.

[25] B. Sahu, D. Mishra, A novel feature selection algorithm using particle swarm optimization for cancer microarray data, Procedia Eng. 38 (2012) 27–31, http://dx.doi.org/10.1016/j.proeng.2012.06.005, International Conference on Modelling Optimization and Computing. URL https://www.sciencedirect.com/science/article/pii/S1877705812019182.

[26] P. Somol, P. Pudil, J. Kittler, Fast branch amp; bound algorithms for optimal feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (7) (2004) 900–912, http://dx.doi.org/10.1109/TPAMI.2004.28.

[27] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39, http://dx.doi.org/10.1007/s10462-009-9124-7.

[28] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[29] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[30] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, J. Artificial Intelligence Res. 11 (1999) 169–198, http://dx.doi.org/10.1613/jair.614.

[31] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) e1249, http://dx.doi.org/10.1002/widm.1249, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249.

[32] S. Nogueira, K. Sechidis, G. Brown, On the stability of feature selection algorithms, J. Mach. Learn. Res. 18 (174) (2018) 1–54, URL http://jmlr.org/papers/v18/17-514.html.

[33] D. Dernoncourt, B. Hanczar, J.-D. Zucker, Analysis of feature selection stability on high dimension and small sample data, Comput. Statist. Data Anal. 71 (2014) 681–693, http://dx.doi.org/10.1016/j.csda.2013.07.012.

[34] Y. Saeys, T. Abeel, Y. Van de Peer, Robust feature selection using ensemble feature selection techniques, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Databases, in: Lecture Notes in Computer Science, Springer, 2008, pp. 313–325, http://dx.doi.org/10.1007/978-3-540-87481-2_21.

[35] V. Bolón-Canedo, A. Alonso-Betanzos, Recent Advances in Ensembles for Feature Selection, Intelligent Systems Reference Library, Springer International Publishing, 2018, URL https://www.springer.com/gp/book/9783319900797.

[36] B. Pes, N. Dessì, M. Angioni, Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data, Inf. Fusion 35 (2017) 132–147, http://dx.doi.org/10.1016/j.inffus.2016.10.001, URL https://www.sciencedirect.com/science/article/pii/S1566253516300847.

[37] Z. He, W. Yu, Stable feature selection for biomarker discovery, Comput. Biol. Chem. 34 (4) (2010) 215–225, http://dx.doi.org/10.1016/j.compbiolchem.2010.07.002.

[38] P. Yang, Y. Hwa Yang, B.B. Zhou, A.Y. Zomaya, A review of ensemble methods in bioinformatics, Curr. Bioinf. 5 (4) (2010) 296–308, http://dx.doi.org/10.2174/157489310794072508.

[39] A. Torrente, M. Lukk, V. Xue, H. Parkinson, J. Rung, A. Brazma, Identification of cancer related genes using a comprehensive map of human gene expression, PLOS ONE 11 (6) (2016) 1–20, http://dx.doi.org/10.1371/journal.pone.0157484.

[40] R.P. Igo Jr., T.G. Kinzy, J.N. Cooke Bailey, Genetic risk scores, Curr. Protoc. Hum. Genet. 104 (1) (2019) e95.

[41] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2010) 392–398, http://dx.doi.org/10.1093/bioinformatics/btp630, URL https://academic.oup.com/bioinformatics/article/26/3/392/213807.

[42] E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination, J. Mach. Learn. Res. 10 (Jul) (2009) 1341–1366.

[43] Y. Xu, Z. Yu, W. Cao, C.L.P. Chen, J. You, Adaptive classifier ensemble method based on spatial perception for high-dimensional data classification, IEEE Trans. Knowl. Data Eng. 33 (7) (2021) 2847–2862, http://dx.doi.org/10.1109/TKDE.2019.2961076.

[44] L. Morán-Fernández, V. Bolón-Canedo, A. Alonso-Betanzos, Centralized vs. distributed feature selection methods based on data complexity measures, Knowl.-Based Syst. 117 (2017) 27–45, http://dx.doi.org/10.1016/j.knosys.2016.09.022, URL https://www.sciencedirect.com/science/article/pii/S0950705116303537 volume, Variety and Velocity in Data Science.

[45] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, A. Alonso-Betanzos, Ensemble feature selection: Homogeneous and heterogeneous approaches, Knowl.-Based Syst. 118 (2017) 124–139, http://dx.doi.org/10.1016/j.knosys.2016.11.017.

[46] P. Křížek, J. Kittler, V. Hlaváč, Improving stability of feature selection methods, in: W.G. Kropatsch, M. Kampel, A. Hanbury (Eds.), Computer Analysis of Images and Patterns, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 929–936.

[47] F. Ferri, P. Pudil, M. Hatef, J. Kittler, Comparative study of techniques for large-scale feature selection, in: E.S. Gelsema, L.S. Kanal (Eds.), Pattern Recognition in Practice IV, 16 of Machine Intelligence and Pattern Recognition, North-Holland, 1994, pp. 403–413, http://dx.doi.org/10.1016/B978-0-444-81892-8.50040-7, URL https://www.sciencedirect.com/science/article/pii/B9780444818928500407.

[48] P. Somol, P. Pudil, Oscillating search algorithms for feature selection, in: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, 2, 2000, pp. 406–409, http://dx.doi.org/10.1109/ICPR.2000.906098, vol. 2.

[49] X. Li, X. Wang, G. Xiao, A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications, Brief. Bioinform. 20 (1) (2017) 178–189, http://dx.doi.org/10.1093/bib/bbx101, arXiv:https://academic.oup.com/bib/article-pdf/20/1/178/27689776/bbx101.pdf.

[50] R. Kolde, S. Laur, P. Adler, J. Vilo, Robust rank aggregation for gene list integration and meta-analysis, Bioinformatics 28 (4) (2012) 573–580, http://dx.doi.org/10.1093/bioinformatics/btr709, arXiv:https://academic.oup.com/bioinformatics/article-pdf/28/4/573/16911737/btr709.pdf.

[51] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, 3, 2003, 523–528. DOI: 10.1109/CSB.2003.1227396.

[52] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238, http://dx.doi.org/10.1109/TPAMI.2005.159.

[53] R.J. Urbanowicz, R.S. Olson, P. Schmitt, M. Meeker, J.H. Moore, Benchmarking relief-based feature selection methods for bioinformatics data mining, J. Biomed. Inform. 85 (2018) 168–188, http://dx.doi.org/10.1016/j.jbi.2018.07.015, URL https://www.sciencedirect.com/science/article/pii/S1532046418301412.

[54] X. Wang, B. Wang, L. Shi, M. Chen, An improved combination feature selection based on relieff and genetic algorithm, in: 2010 5th International Conference on Computer Science Education, 2010, pp. 1340–1343, http://dx.doi.org/10.1109/ICCSE.2010.5593712.

[55] Y. Zhang, C. Ding, T. Li, Gene selection algorithm by combining reliefF and mRMR, BMC Genomics 9 Suppl 2 (2008) S27.

[56] Y. Zhang, C. Ding, T. Li, Gene selection algor ithm by combining relieff and mrmr, BMC Genomics 9 Suppl 2 (2008) S27, http://dx.doi.org/10.1186/1471-2164-9-S2-S27.

[57] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, J. R. Stat. Soc. Ser. B Stat. Methodol. 73 (3) (2011) 273–282, URL http://www.jstor.org/stable/41262671.

[58] Q. Hu, P. Zeng, L. Lin, The dual and degrees of freedom of linearly constrained generalized lasso, Comput. Statist. Data Anal. 86 (2015) 13–26, http://dx.doi.org/10.1016/j.csda.2014.12.010, URL https://www.sciencedirect.com/science/article/pii/S0167947314003582.

[59] E.L. de Maturana, Y. Ye, M.L. Calle, N. Rothman, V. Urrea, M. Kogevinas, S. Petrus, S.J. Chanock, A. Tardón, M. García-Closas, A. González-Neira, G. Vellalta, A. Carrato, A. Navarro, B. Lorente-Galdós, D.T. Silverman, F.X. Real, X. Wu, N. Malats, Application of multi-snp approaches bayesian lasso and auc-rf to detect main effects of inflammatory-gene variants associated with bladder cancer risk, PLOS ONE 8 (12) (2014) 1–11, http://dx.doi.org/10.1371/journal.pone.0083745.

[60] S. Zheng, W. Liu, An experimental comparison of gene selection by lasso and dantzig selector for cancer classification, Comput. Biol. Med. 41 (11) (2011) 1033–1040, http://dx.doi.org/10.1016/j.compbiomed.2011.08.011, URL https://www.sciencedirect.com/science/article/pii/S0010482511001879.

[61] Z.Y. Algamal, M.H. Lee, Penalized logistic regression with the adaptive lasso for gene selection in high-dimensional cancer classification, Expert Syst. Appl. 42 (23) (2015) 9326–9332, http://dx.doi.org/10.1016/j.eswa.2015.08.016, URL https://www.sciencedirect.com/science/article/pii/S0957417415005618.

[62] Z. Li, W. Xie, T. Liu, Efficient feature selection and classification for microarray data, PLOS ONE 13 (8) (2018) 1–21, http://dx.doi.org/10.1371/journal.pone.0202167.

[63] H. Sanz, C. Valim, E. Vegas, J.M. Oller, F. Reverter, SVM-RFE: selection and visualization of the most relevant features through non-linear kernels, BMC Bioinformatics 19 (1) (2018) 432.

[64] S. Mishra, D. Mishra, Svm-bt-rfe: An improved gene selection framework using bayesian t-test embedded in support vector machine (recursive feature elimination) algorithm, Karbala Int. J. Mod. Sci. 1 (2) (2015) 86–96, http://dx.doi.org/10.1016/j.kijoms.2015.10.002, URL https://www.sciencedirect.com/science/article/pii/S2405609X15300671.

[65] J. Arlandis, J. Perez-Cortes, J. Cano, Rejection strategies and confidence measures for a k-nn classifier in an ocr task, in: 2002 International Conference on Pattern Recognition, Vol. 1, 2002, pp. 576–579, http://dx.doi.org/10.1109/ICPR.2002.1044806, vol. 1.

[66] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[67] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.

[68] L. Kirkland, F. Kanfer, S. Millard, Lasso tuning parameter selection, in: Annual Proceedings of the South African Statistical Association Conference: Proceedings of the 57th Annual Conference of the South African Statistical Association for 2015 (SASA 2015), 2015, pp. 49–56.