Engineering Applications of Artificial Intelligence 70 (2018) 92–108

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A data mining framework based on boundary-points for gene selection from DNA-microarrays: Pancreatic Ductal Adenocarcinoma as a case study

Juan Ramos[a], José A. Castellanos-Garzón[a,b,*], Juan F. de Paz[a], Juan M. Corchado[a,c]

[a] University of Salamanca, IBSAL/BISITE Research Group, Edificio I+D+i, 37007 Salamanca, Spain [1]
[b] University of Coimbra, CISUC, ECOS Research Group, Pólo II - Pinhal de Marrocos, 3030-290 Coimbra, Portugal [2]
[c] University of Salamanca, Osaka Institute of Technology, BISITE Research Group, Edificio I+D+i, 37007 Salamanca, Spain

ARTICLE INFO

Keywords:
Feature selection
Gene selection
Data mining
Cluster analysis
Evolutionary computation
Boundary point
DNA-microarray
Visual analytics
Filter method
Boundary gene

ABSTRACT

Gene selection (or feature selection) from DNA-microarray data can be focused on different techniques, which generally involve statistical tests, data mining and machine learning. In recent years there has been an increasing interest in using hybrid-technique sets to face the problem of meaningful gene selection; nevertheless, this issue remains a challenge. In an effort to address the situation, this paper proposes a novel hybrid framework based on data mining techniques and tuned to select gene subsets, which are meaningfully related to the target disease conducted in DNA-microarray experiments. For this purpose, the framework above deals with approaches such as statistical significance tests, cluster analysis, evolutionary computation, visual analytics and boundary points. The latter is the core technique of our proposal, allowing the framework to define two methods of gene selection. Another novelty of this work is the inclusion of the age of patients as an additional factor in our analysis, which can leading to gaining more insight into the disease. In fact, the results reached in this research have been very promising and have shown their biological validity. Hence, our proposal has resulted in a methodology that can be followed in the gene selection process from DNA-microarray data.

1. Introduction

Advances in *bioinformatics* in the last years have made it possible to apply *artificial intelligence* hybrid techniques to further understand and validate the achieved results. Bioinformatics is in fact one of the most controversial areas of research at present, since it deals with the development and/or application of methods and algorithms to turn biological data into knowledge of biological systems, often requiring further experimentation from initial data, Bourne and Wissig (2003).

Meanwhile, *data mining* and *functional genomics* have also gained attention since the publication of several complete genome sequences as well as the human genome. One of the most advanced and challenging ways of studying molecular events has been the monitoring of gene expression patterns from *DNA-microarrays*. Microarrays can be viewed as a type of device (a chip) in which, a large number of diverse entities, such as peptides, oligonucleotides, biological molecules, cells, tissues, etc., are located on its surface, and placed in an orderly and accurate way. Once these entities are attached on the surface of the chip, they can be simultaneously evaluated in a single essay (Berrar et al., 2003; Chan and Kasabov, 2004; Geoffrey et al., 2004; Jiang et al., 2004; Speed, 2003).

An important research area developed from the data domain above is *gene/feature selection*, which deals with the discovery of gene subsets relevant for a particular target. Such genes are called *informative* (or *differentially expressed genes*) and are the basis for developing *classifiers* in the study of disease diagnosis and prognosis. They are also studied by pharmaceutical companies, whose efforts are focused on identifying those genes that can be targeted by drugs (Inza et al., 2004; Jager et al., 2003; Kumari and Swarnkar, 2011; Lazar et al., 2012; Simeka et al., 2004). While significant efforts have been placed in the development of new methods and strategies to discover informative genes, the problem remains a challenge today since there is not a single technique able to solve all the underlying issues. In general aspects, feature selection methods can be classified into four categories: *filters, wrappers, embedded* and a more recent method group known as *ensemble* (Natarajan and Ravi, 2014; Shraddha et al., 2014; Tyagi and Mishra, 2013; Wang et al., 2005). Each of these categories demanding unification of different techniques as *supervised* and *unsupervised learning, evolutionary computation,*

* Corresponding author at: University of Salamanca, IBSAL/BISITE Research Group, Edificio I+D+i, 37007 Salamanca, Spain .
E-mail addresses: juanrg@usal.es (J. Ramos), jantonio@usal.es (J.A. Castellanos-Garzón), fcofds@usal.es (J.F. de Paz), corchado@usal.es (J.M. Corchado).

[1] http://bisite.usal.es.

[2] https://www.cisuc.uc.pt/groups/show/ecos.

https://doi.org/10.1016/j.engappai.2018.01.007
Received 2 June 2016; Received in revised form 30 September 2017; Accepted 20 January 2018

visual analytics, among others, in order to gain insight into the problem at hand.

Hence, this research proposes a framework relating hybrid techniques of artificial intelligence and statistics to gene subset selection from gene expression data, which we call *HybridFrame*. Three major characteristics can be stressed from HybridFrame. To begin, it develops a methodology addressing two different methods of gene selection, one based on evolutionary algorithms and the other one, based on the intersection of results coming from different methods. Secondly, the core idea of HybridFrame has been focused on cluster boundary genes to determine informative genes. Furthermore, this framework suitably links a set of hybrid techniques as statistical significance tests, cluster analysis, genetic algorithms, visual analytics and boundary points, to successively reduce (as a filtering strategy) the involved dataset until reaching a small subset of meaningful genes related to the target disease.

We have used hybrid techniques to build a data mining framework for gene selection tasks, because they provide more robust and stable solutions than simple methods (Guyon, 2003; Jager et al., 2003; Lazar et al., 2012). Generally, simple methods of gene selection assume that some criterion should be met in data, which does not have to be true for all data types. Hence, hybrid techniques fusion different simple methods to reach solutions holding more than one criterion, making solutions more stable with respect to variations in data. On the other hand, hybrid techniques are more flexible to changes in user needs and allow us to replace the methods taking place in the overall proposal without carrying out meaningful changes.

1.1. Case study, impact and motivation

As a case study to apply and validate our proposal, we have focused our attention on the tissue sample study of *pancreatic ductal adenocarcinoma* (PDAC) through microarray technology, given that PDAC has been identified as one of the most aggressive types of existing cancer (Badea et al., 2008a, b), with a majority of cases, unfortunately, detected in advanced stages due to the lack of early symptoms, Crnogorac-Jurcevic et al. (2013). Hence, PDAC patients have a median survival of less than six months and a five-year survival rate of about 5% patients, Hezel et al. (2006). Indeed, 60%–70% patients already present metastasis when the cancer is detected. In spite of the fact that much knowledge from PDAC molecular processes has been revealed in the last few years, the scientific community is still far from developing effective therapies leading to an ability to face this pathology.

One of the main causes for this is the drug's low effectiveness in PDAC treatment, which has been attributed to a high dynamic relation between cancer cells and the stroma, Bhaw-Luximon and Jhurry (2015). This has resulted in many events allowing stroma formation to act as a protective environment of the tumor. Moreover, unlike other influential factors such as alcoholism, previous lesions, smoking or genetic issues, age appears to be especially important in PDAC. Every cancer has a strong relation to age due to several cellular processes, as is the case of senescence, but for PDAC, this relation appears to be more remarkable than other cancers. In fact, 85% of pancreatic cancer cases involve patients older than 65-years old with a diagnosis mean age of 73-years old, Koorstra et al. (2008). For that reason, this research introduces the age factor for further analysis of its influence in cancer patients. Hence, the goal of our proposal with the current case study is to identify age-related gene subsets, which may influence the severity of the disease. In that sense, such genes apart of being age-related, should also be able to capture the greatest variations of their expression levels (relation qualitative + quantitative).

Finally, to reach all goals proposed in this research, the remaining sections of this paper have been divided as follows: Section 2 describes works related to the feature selection process and our proposal. Section 3 develops the framework for gene selection and explains each of its components as their interactions. Section 4 describes the dataset to be used, experiments, results and discussion after applying an implementation of the introduced framework. Section 5 presents the conclusions of this paper whereas Appendix outlines a set of visualizations supporting the results. References used in this research have been given as the final part of this paper.

2. Related work

Feature selection (FS) can be generically defined as the process of extracting feature or gene subsets whose expression level values are representative of a particular target feature, i.e., clinical or biological annotation (Inza et al., 2004; Jager et al., 2003; Kumari and Swarnkar, 2011; Lazar et al., 2012). FS is a very active research area in the analysis of *gene expression microarray*, which is contributing to the development of the field as a result of involved data mining and machine learning techniques, TunedIT (2008). Particularly, FS from microarrays is addressed to identify/discover those genes which are expressed differentially according to a determined target disease (namely, *informative genes*). As previously stated in the introduction, there is a large number of approaches in the literature dealing with this issue and with potential application in the area of disease prediction and discovery, gene regulatory network reconstruction, pharmaceutical industry, among others (Golub et al., 1999; Penfold and Wild, 2011). However, the many challenges posed by this research field require new approaches.

Due to the wide range of papers proposed to face the FS problem in microarrays and facilitate the study of the area, FS methods have been divided into the following four categories: *filters, wrappers, embedded* and *ensemble*. Filter methods have been directed to discriminate or filter features/genes based on the intrinsic properties of the dataset by estimating their relevance scores to state a cut-off schema where an upper/lower bound is imposed in order to choose features with the best scores. According to Guyon (2003) and Lazar et al. (2012), this scheme could favor gene identification to be targeted pharmaceutically. Wrapper methods use a classifier to find the most discriminant feature subset by minimizing an error prediction function (Ambroise and McLachlan, 2002; Díaz-Uriarte and Alvarez, 2006; Ruiz et al., 2006; Yee et al., 2005; Zhou and Tuck, 2007). These methods tend to consume a lot of runtime and their results depend on the type of used classifier. Embedded methods are similar to wrapper, but allow the learning method to interact, which reduces the runtime taken by wrapper methods (Efron et al., 2004; Hernandez et al., 2007; Lazar et al., 2012; Quinlan, 1994; Saeys et al., 2007). Ensemble methods are relatively new and recombine results from different FS techniques to achieve a more stable feature subset, since small perturbations in the training set can have effects on the results of a FS method applied individually (Haury et al., 2011; Moorthy and Saberi, 2012; Nguyen et al., 2015). Therefore, ensemble methods come to face the instability difficulty presented by some of the approaches previously explained.

Since the FS methodology followed by the proposed framework is based on a filtering strategy to successively reduce a dataset until the target gene subset has been achieved, we are going to focus our attention on some of the main features presented by filter techniques. This will allow us to highlight two trends followed by filter methods. The first type refers to methods selecting the top ranking features, which are based on the relevance value assignment to each feature/gene (*ranking methods*). The relevance value estimation is carried out by a scoring function preselected according to the pursued target (Jaeger et al., 2003; Liu et al., 2005; Peddada et al., 2003; Yang et al., 2006). The second type of trend includes *space search methods*, which are engaged to optimize an objective function by generally involving maximum relevance and minimum noise for the found gene subsets (Ding and Peng, 2005; Mohamed et al., 2015; Wang et al., 2005; Xing et al., 2001). According to the classifications above, those of Lazar et al. (2012) have stated a taxonomy for FS methods as follows: Raking methods can be classified as either *univariate* or *bivariate* (Deng et al., 2004; Long et al., 2001; Thomas et al., 2001; Tusher et al., 2001). Univariate methods can be further classified as either *parametric* or *non-parametric* whereas bivariate methods can be *greedy* or *all-pairs* (Bo and Jonassen, 2002; Geman et al., 2003; Yeung and Bumgarner, 2003). For its part, space search methods are *multivariate* (Ding and Peng, 2005; Wang et al., 2005; Xing et al., 2001).

In general aspects, ranking methods aim to select the top scoring features/genes by discriminating the rest in a four-staged approach: (1) select a scoring function that assigns a score to each feature and sort the whole dataset based on each score. (2) estimate the statistical significance of the assigned scores (i.e., p-values). (3) select the top ranking features according to the two previous stages and (4), validate the gene subset found. Unlike the approach above, space search methods optimize the combination of significance and redundance (the least redundance) to select meaningful gene subsets by following three steps, which consist of building an objective function to optimize, defining the search algorithm for gene subsets by using the objective function, and a validity process of the solution. The latter is a common point given in both approaches (ranking and space search method), and is mandatory for any FS method. The validity (or evaluation) process of a gene subset is usually called *signature* and if the goals are aimed at classifying the disease type, then the gene subset is evaluated according to the accuracy of a determined classifier. In contrast, if the goals are focused on identification of biomarkers for further research, then the genes found are validated separately with respect to the statistical significance of their assigned scores (Lazar et al., 2012; Natarajan and Ravi, 2014; Shraddha et al., 2014; Tyagi and Mishra, 2013; Wang et al., 2005).

To conclude this section, we can say that after reviewing the previous literature, filter methods have widely been used in the FS process complex. They have also been integrated into more complex systems, coupling machine learning and/or data mining techniques, for which good results have been achieved. On the other hand, the application of a single standard method to find informative genes, i.e., assigning relevance indices to genes by using some of the given statistical tests and then, ranking them to select the top k genes is not the best option since they are often highly correlated, Jager et al. (2003). Hence, we propose a data mining framework linking different techniques (as previously explained) to select informative gene subsets. Unlike all approaches presented in this section, our proposal brings a strategy combining cluster analysis with boundary gene-points (Castellanos-Garzón, 2012; Castellanos-Garzón et al., 2013; Castellanos-Garzón and Díaz, 2013).

3. One data mining framework, two methods of gene selection

This section describes components, methods and methodology followed by the HybridFrame framework to select gene subsets being meaningful for the target data domain. Since HybridFrame is based on data mining techniques, we have focused our efforts on the combination of areas such as evolutionary computation, visual analytics, and cluster analysis, among others to develop a methodology to follow in the domain of gene expression data.

Consequently, we stress the fact that the use of data mining involves a variety of data analysis tools to discover patterns and relations from data in a way that may be used to make valid predictions (Han and Kamber, 2006; Jiang et al., 2004; Olson and Delen, 2008). In accordance with the above, classifying DNA-microarray data according to their similarity degree is one of the main goals of data mining, since the organization of objects in affinity groups is one way of discovering knowledge (Jain and Dubes, 1998; Kaufman and Rousseeuw, 2005). Hence, cluster analysis can be intended as one component of exploratory data analysis, which means sifting through data to make sense out of measurements using whatever means are available, whereas the use of evolutionary computation develops blind search methods, inspired by natural selection mechanisms, and lead to solving complex optimization problems (Goldberg, 1989; Holland, 1992). Finally, this framework introduces visual analytics for aggregating, summarizing and visualizing information generated during interactive cluster analysis (Keim, 2002; Schroeder et al., 2001). According to all the above, Fig. 1 displays a diagram representing all processes performed by HybridFrame, which will be explained as modules (and methods) in the following subsections. Finally, a version of HybridFrame has been implemented by joining programming languages *R-Project* (R Core Team, 2015), using packages *clustergas* and *hybridHclust* (Castellanos-Garzón and Díaz, 2012; Chipman et al., 2006), *Java* and *Java-3D*.

3.1. Statistical filtering module (SFM)

According to Fig. 1, SFM is the first module to run after selecting a target dataset from a data repository. This module is responsible for a preliminary data processing and the first gene filtering processes based on statistical significance. Thus, the first process in this module consists of a data treatment by removing control probes, standardizing (as for example, normalizing data to mean 0 and variance 1), and applying algorithms of missing data treatment if needed. Next, two filtering processes based on gene significance are followed. The aim is to successively reduce the input dataset (result #1) from different statistical significance criteria. The first applied filter method is the Mann–Whitney test (Weiss, 2005) which involves a non-parametric test, since we assume the data do not belong to any particular distribution. This test states a null hypothesis claiming that samples (genes) come from the same population, whereas the alternative hypothesis claims that samples come from different populations, i.e., a population has bigger values than the other one. The goal of applying this test is to reject the null hypothesis to filter genes belonging to different populations, which is meaningful for the study. Hence, such genes present a low *p-value* (high significance), consequently having a high probability of being related to the target disease. The end step of this method is to select a p-value cutoff to filter out genes with the highest significance, which has been fixed in 0.05, i.e., genes with p-value < 0.05 are selected. Finally, a reduced dataset is returned to the following filter method in SMF.

The next step after applying the Mann–Whitney test is to select the second filter method in relation to user goals. In this case, the module implemented five filter methods, although new methods can be added. We are going to explain the Kruskal–Wallis test since it was used in our case study. The remaining filter methods can be found in the given literature; they are specifically referenced in Berrar et al. (2003) and Lazar et al. (2012). The Kruskal–Wallis test (McDonald, 2014) is also a non-parametric test to determine whether the mean ranks of the groups are the same as the null hypothesis. For practical proposes, this test is equivalent to a variance analysis (ANOVA), although it replaces data by categories; moreover, it is an extension of the Mann–Whitney test for three or more groups. This test can be used by the users when they introduce an variable measurement external (such as the variable *age* introduced in our case study) to the dataset to filter out genes related to the variable of interest. Thus, the goal of this test is to extract those genes with higher significance with regard to an external variable, which is translated to select genes with low p-values (rejecting the null hypothesis). In this case, genes associated with a p-value < 0.05 are passed to the next phase of the framework. In consequence, this filter method returns a reduced dataset. In the absence of an external variable, the user can select one of the remaining filter methods offered by SFM to reduce the dataset accordingly. The result of the second chosen filter method will be the end result given by this module to the next HCMM module.

Finally, we want to stress that although the Mann–Whitney test has been prefixed as the first used filter method, it can be replaced by another method available from the filter method repository of the module. The Mann–Whitney test has been selected because it is a *ranking method* and one of the most commonly used methods in gene selection, Lazar et al. (2012). Also note that we can only use at most two linked filter methods in the current module, before passing to the next module.

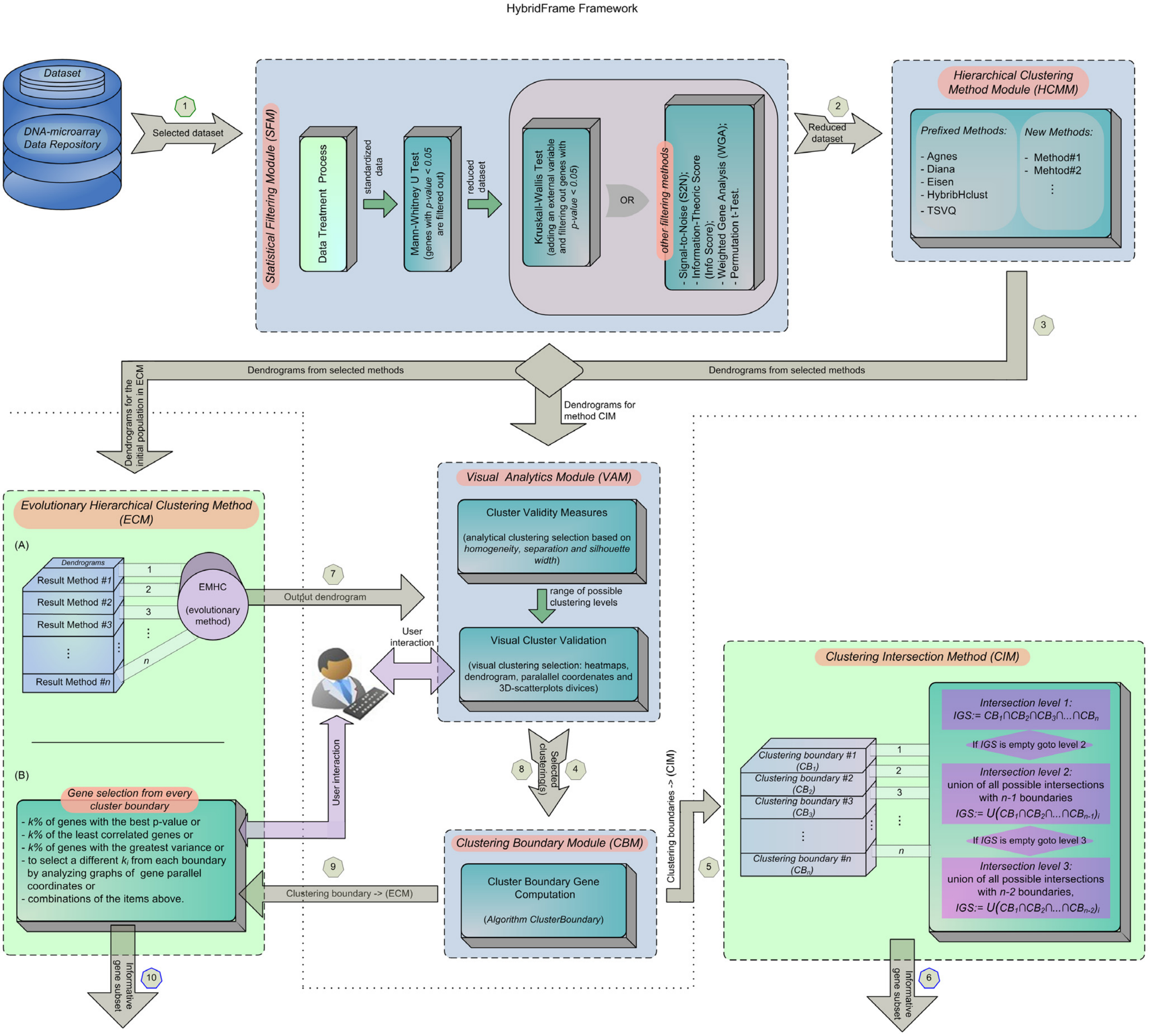
HybridFrame Framework
Dataset
DNA-microarray Data Repository
1
Selected dataset
Statistical Filtering Module (SFM)
Data Treatment Process
standardized data
Mann-Whitney U Test (genes with p-value < 0.05 are filtered out)
reduced dataset
Kruskall-Wallis Test (adding an external variable and filtering out genes with p-value < 0.05)
OR
other filtering methods
- Signal-to-Noise (S2N);
- Information-Theoric Score (Info Score);
- Weighted Gene Analysis (WGA);
- Permutation t-Test.
2
Reduced dataset
Hierarchical Clustering Method Module (HCMM)
Prefixed Methods:
- Agnes
- Diana
- Eisen
- HybribHclust
- TSVQ
New Methods:
- Method#1
- Method#2
⋮
3
Dendrograms from selected methods
Dendrograms from selected methods
Dendrograms for the initial population in ECM
Dendrograms for method CIM
Evolutionary Hierarchical Clustering Method (ECM)
(A)
Dendrograms
Result Method #1
Result Method #2
Result Method #3
⋮
Result Method #n
1
2
3
⋮
n
EMHC
(evolutionary method)
7
Output dendrogram
User interaction
(B)
Gene selection from every cluster boundary
- k% of genes with the best p-value or
- k% of the least correlated genes or
- k% of genes with the greatest variance or
- to select a different k_i from each boundary by analyzing graphs of gene parallel coordinates or
- combinations of the items above.
User interaction
9
Clustering boundary -> (ECM)
10
Informative gene subset
Visual Analytics Module (VAM)
Cluster Validity Measures
(analytical clustering selection based on homogeneity, separation and silhouette width)
range of possible clustering levels
Visual Cluster Validation
(visual clustering selection: heatmaps, dendrogram, paralalel coordenates and 3D-scatterplots divices)
8
Selected clustering(s)
4
Clustering Boundary Module (CBM)
Cluster Boundary Gene Computation
(Algorithm ClusterBoundary)
Clustering boundaries -> (CIM)
5
Clustering Intersection Method (CIM)
Clustering boundary #1 (CB_1)
Clustering boundary #2 (CB_2)
Clustering boundary #3 (CB_3)
⋮
Clustering boundary #n (CB_n)
1
2
3
⋮
n
Intersection level 1:
IGS:= CB_1∩CB_2∩CB_3∩...∩CB_n
If IGS is empty goto level 2
Intersection level 2:
union of all possible intersections with n-1 boundaries
IGS:= U(CB_1∩CB_2∩...∩CB_n-1)_i
If IGS is empty goto level 3
Intersection level 3:
union of all possible intersections with n-2 boundaries,
IGS:= U(CB_1∩CB_2∩...∩CB_n-2)_i
6
Informative gene subset

Fig. 1. Chart representing data mining framework HybridFrame for gene selection. This consists of four modules (SFM, HCMM, VAM and CBM) and two gene selection methods (ECM and CIM), all of which have been linked through their results. In this case, the results of each phase are identified by arrows and their corresponding order number. The initial and end results (results 1, 6 and 10) have been underlined in different colors unlike the intermediate results.

We have found that at least two methods are necessary to reduce enough a big dataset. Nevertheless, the user can use a single filter method if it is enough. The idea pursued by SFM is to reduce the dataset as much as possible by removing genes assumed as noise, before running the module for cluster analysis, i.e., HCMM.

3.2. Hierarchical clustering method module (HCMM)

The goal of this module is to partition the dataset resulting from the module above into subsets (clusters) in order to move the gene selection task from the whole current dataset to smaller gene subsets. The idea consists of applying data clustering methods to divide the complex task of gene selection from a big dataset into small subsets (*divide and conquer strategy*), identified by their gene similarity. Although this module does not really perform a gene filtering task, it partitions the data for the following stages. This is an open module in the sense that new clustering methods can be added to its method repository. Furthermore, there is a subset of prefixed methods, which has been implemented within the module. Note that these methods render hierarchical clusterings, which are of great importance in the analysis of biological data, Jain and Dubes (1998). They are also the most commonly used methods in the DNA-microarray data domain, each one performing a different clustering strategy.

In particular, the *Agnes* algorithm builds a hierarchy of clusterings, Kaufman and Rousseeuw (2005). At first, each data is a small cluster by itself. Clusters are merged until only one large cluster containing all the data remains. At each stage the two nearest clusters are combined to form one larger cluster. The *Diana* algorithm performs the task in reverse order, starting from one large cluster containing all data, Kaufman and Rousseeuw (2005). Clusters are divided until each cluster contains only a single piece of data. At each stage, the cluster with the largest diameter is selected to be split. The *Eisen* algorithm carries out an agglomerative hierarchical clustering in which each cluster is represented by the mean vector for data in the cluster, Eisen et al. (1998). The *TSVQ* algorithm builds a divisive hierarchical clustering, so the data must be subdivided recursively into two clusters, Macnaughton-Smith et al. (1965). Hence, 2-means is used to find a subdivision. The *HybridHclust* algorithm is a divisive hierarchical clustering where TSVQ is applied to data with the constraint that mutual clusters cannot be divided, Chipman et al. (2006). Within each mutual cluster, TSVQ is re-applied to render a top-down hybrid in which a mutual cluster structure is retained. Since HybridHclust is based on TSVQ, it implicitly uses squared Euclidean distance between data.

As a first step to run in this module, the user must specify the clustering methods and settings to be used in the cluster analysis of the input dataset. After that, selected methods are run on the current dataset and the result-dendrograms of each method are given as output to the following stages.

3.3. Visual analytics module (VAM)

The goal of this module (VAM, Fig. 1) is to select the most suitable clustering (high quality clustering) from each input dendrogram to compute its boundary points in the next module. Hence, VAM consists of two parts, analytical and visual. In the analytical part, internal measures of cluster validity are applied to input dendrograms to estimate level ranges (or intervals) with high quality clusterings. This process is responsible for computing an interval with the level numbers of the best clusterings for each input dendrogram. To do this, this module relies on *homogeneity, separation* and *silhouette width* measures (Jiang et al., 2004; Kaufman and Rousseeuw, 2005), which are applied to each level of a dendrogram to select the one with the best score for each one of the measures. Of the three level numbers given by the used measures, the two highest numbers are selected to be part of the lower and upper extremes and to create the level interval with the clusterings to be analyzed. In this way, the level interval for each dendrogram is computed and returned to the next process in this module.

The task of the second part of VAM (i.e., the process of visual clustering validity) consists of choosing and visually validating a level (clustering) from each level interval computed in the process above. For this propose, each dendrogram is explored from its level interval through a linked visualization set, supporting heatmaps, dendrograms, parallel coordinates, 3D-scatterplots and boundary gene visualizations, as shown in Appendix. Once this process is completed, a clustering for each used method is returned to the following module of the framework. Note that this module has an additional input of a dendrogram coming from method ECM as well as an user interaction process, which will allow visually selecting a single clustering for each dendrogram.

3.4. Clustering boundary module (CBM)

The goal of this module (CBM, Fig. 1) is to carry out a filtering process by extracting out the boundary genes for each cluster given from input clusterings to the module. Then, for each input clustering, CBM computes the boundary genes of each cluster to return a new clustering (called *clustering boundary*) whose clusters only have their boundary genes. The boundary point algorithm used for this purpose is focused on the *ClusterBoundary algorithm* given in Castellanos-Garzón et al. (2013). In this case, we introduce *principal component analysis* (Jolliffe, 2002) to the algorithm to reduce the data dimension with the aim of minimizing the number of points/genes in the clustering boundary. In general aspects, ClusterBoundary is based on the boundary concept related to *metric spaces*, Namely, the set of points in the closure of a cluster that do not belong to the interior of the cluster.

We stress the fact that boundary points are data located at the region margin of densely distributed points. In the case of a cluster, boundary points become representative of each cluster given from a clustering. Therefore, they are able to summarize part of the information provided by a cluster and thus discriminate the remaining points in the cluster. Hence, cluster boundary genes are good candidates to be representative of differently expressed genes. To conclude this module, once the boundary for each input clustering has been computed, the result is targeted to the two methods responsible for finding informative gene subsets.

3.5. Clustering intersection method (CIM)

As previously stated, this framework consists of two different methods to discover informative genes and a set of core modules, which are run before linking these methods. In this case, the CIM method is based on the idea of boundary intersections coming from different clustering methods. The hypothesis pursued in this approach is that *boundary genes achieved from the intersection of different clustering boundaries coming from different methods, which develop different cluster strategies on data, are the main candidates to be informative genes*. Under this principle, we have developed an algorithm generalizing the CIM method given in Fig. 1. The algorithm basically states all possible intersection levels between the clustering boundaries. That is, in level 1, it intersects all input clustering boundaries (say, n boundaries in form of n sets, boundary sets) to capture all boundary genes repeated. The method ends if the resulting gene set is nonempty; otherwise, intersection level 2 is applied. This intersection then computes the union of all possible intersections formed by $n - 1$ input boundaries. If the resulting gene set is nonempty, then the method ends. Otherwise, intersection level 3 is applied as one level 2, but in this case, taking all possible $n - 2$ input boundaries to join their intersections. The process above is repeated for the next levels until an intersection be nonempty or the level number reaches value $n - 1$, i.e., all combinations of 2 sets taken from n sets. If intersection union in level $n - 1$ is empty, which would be very rare, then the boundary sets are disjoint and their union is computed since all genes are significative for the method. Note that running this method involves the sequence of results $\langle 1 - 2 - 3 - 4 - 5 - 6 \rangle$ followed in Fig. 1. Finally, the formal algorithm of CIM has been given below:

Algorithm CIM (Algorithm 1)
Input: A set of clustering boundaries $\mathfrak{I} = \{CB_1, CB_2, \cdots, CB_n\}$
Output: IG, an informative gene set.

```
$\mathfrak{B} := \emptyset$;
for all $CB$ in $\mathfrak{I}$ do
    % Computing the union of all clusters for each $CB$.
    % Converting clustering boundaries to sets and adding
      them to $\mathfrak{B}$.
    Add($\mathfrak{B}$, $\bigcup_{i=1}^{|CB|} C_i$), where each $C_i$ is a cluster boundary of $CB$;
endfor
% Computing intersection level 1.
$IG := \bigcap_{i=1}^{n} F_i,\ F_i \in \mathfrak{B}$;
$l = 2$; % Starting the loop with intersection level 2.
while $IG \neq \emptyset$ and $l < n$ do
    % Computing intersection level $l$.
    % Computing the union of all possible intersections
      with $n - l + 1$ sets taken from $\mathfrak{B}$.
    $IG := \bigcup_{i=1}^{\binom{n}{n-l+1}} \left( \bigcap_{j=1}^{n-l+1} F_{ij} \right),\ F_{ij} \in \mathfrak{B}$;
    $l = l + 1$;
endwhile
if $IG = \emptyset$ then
    % Computing the union of all boundary sets.
    % At this point all sets are disjoint so that, all genes are
      important.
    $IG := \bigcup_{i=1}^{n} F_i,\ F_i \in \mathfrak{B}$;
endif
end.
```

3.6. Evolutionary hierarchical clustering method (ECM)

The second method to discover informative genes in this framework is ECM as shown in Fig. 1. This method is based on the evolutionary model for clustering EMHC given in Castellanos-Garzón (2012) and Castellanos-Garzón and Díaz (2013). In model EMHC, a set of parameters can be pre-fitted based on specific criteria in order to obtain a concrete clustering method able to adapt to the analyzed problem. However, by varying those parameters, we may possibly achieve a different method. Such an approach is possible through evolutionary computation. Thus, the ECM method is a specific implementation of EMHC, designed to adapt to the case study.

As with all genetic algorithms, ECM starts from an initial population, which in this case consists of dendrograms given as solutions to other methods. The goal of this approach is to improve such solutions based on the evolutionary force of ECM, while high cluster structures captured by other methods are retained by ECM from generation to generation. In this sense, our hypothesis to discover informative genes from this approach is that, *since dendrograms given as ECM solutions inherit, alter, recombine and even improve part of the genetic code (high quality clusters) of good solutions given by others methods, then it is expected that genes located on the boundary of such clusters are strong candidates to be informative genes.*

In order to reach the goal above, ECM has been divided into two parts, where Part-(A) is responsible for running the genetic algorithm of ECM from input dendrograms returned by module HCMM and giving the result-dendrogram to module VAM for its processing. Part-(B) is responsible for choosing an informative gene subset from the clustering boundary (as its input) coming from modules VAM and CBM, which have already processed the result of Part-(A). Note that Part-(B) also includes a process of user interaction dedicated to choose the gene selection strategy for each cluster boundary. Moreover, the running of ECM implies the following sequence of results according to Fig. 1, $\langle 1-2-3-7-8-9-10\rangle$.

3.6.1. Search for pareto optimal solutions

As previously explained, the ECM approach modifies the original one, mainly in the fitness function used to guide the search. In this case, we separate the objectives of the fitness function given in Castellanos-Garzón and Díaz (2013) and based on the concept of *Pareto optimality*, Haupt and Haupt (2004). There is a set of optimal solutions, known as Pareto optimal solutions, non-inferior solutions, or effective solutions. Without additional information, all these solutions are equally satisfactory. The goal is then to find as many of these solutions as possible. If reallocation of resources cannot improve one cost without raising another, then the solution is Pareto optimal. The difficulty of considering all the objectives together is that we rarely find a situation where a single vector represents the optimum solution for all the objectives. A formal definition of Pareto optimality dealing with the minimization problem is as follows, Fonseca and Fleming (1995): a decision vector $\vec{x}^*$ is called Pareto optimal if and only if there is no $\vec{x}$ that dominates $\vec{x}^*$, i.e., there is no $\vec{x}$ such that:

$$\forall i \in [1,k], f_i(\vec{x}) \leq f_i(\vec{x}^*) \text{ and } \exists i \in [1,k], \text{where } f_i(\vec{x}) < f_i(\vec{x}^*).$$

A solution $\vec{x}^*$ strongly dominates a solution $\vec{x}$ if $\vec{x}^*$ is strictly better than $\vec{x}$ in all the objectives. Thus, multi-objective optimization is interested in obtaining a set of non-dominated solutions. Once the concept of Pareto optimality has been shown, we can introduce the modification presented in ECM to be a Pareto evolutionary algorithm. To do so, we transform the fitness functions for dendrogram and clustering, which are:

$$f_d(\mathfrak{G}) = \frac{1}{|\mathfrak{G}|-1}\sum_{i=1}^{|\mathfrak{G}|-1} f_c(\mathfrak{C}_i), \tag{1}$$

where $\mathfrak{G}$ is a dendrogram, $\mathfrak{C}_i$ is the clustering of level i in $\mathfrak{G}$ and f_c is the recurrent fitness function to evaluate a clustering of $\mathfrak{G}$, which is defined as,

$$f_c(\mathfrak{C}_{i+1}) = \frac{\mathcal{S}_1^*(\mathfrak{C}_{i+1})}{g-k+1} - \frac{\mathcal{H}_1^*(\mathfrak{C}_{i+1})}{k-1} + \max\mathfrak{D}, \tag{2}$$

where $\mathcal{S}_1^*(\mathfrak{C}_{i+1})$ and $\mathcal{H}_1^*(\mathfrak{C}_{i+1})$ are separation and homogeneity for clustering $\mathfrak{C}_{i+1}$ respectively, being defined in Castellanos-Garzón et al. (2013), $k = |\mathfrak{C}_i|$ and $g = \binom{k}{2}$ is the number of distances among the clusters of $\mathfrak{C}_{i+1}$. $\max\mathfrak{D}$ is the maximum distance from proximity matrix $\mathfrak{D}$ of the current dataset. Note that the problem stated by both functions is one of maximization to achieve high quality dendrograms. But now, these fitness functions can be redefined as a vector of two objective-components measuring separation and homogeneity separately, i.e.:

$$f_d^*(\mathfrak{G}) = \langle \mathcal{S}(\mathfrak{G}), \max\mathfrak{D} - \mathcal{H}(\mathfrak{G})\rangle, \tag{3}$$

where $\mathcal{S}$ and $\mathcal{H}$ are measures of separation and homogeneity respectively defined for dendrograms. Then, the goal is to maximize the two components of f_d^*. On the other hand, $\mathcal{S}$ and $\mathcal{H}$ have been defined in relation to clusterings $\mathfrak{C}_i$ of $\mathfrak{G}$ as:

$$\mathcal{S}(\mathfrak{G}) = \frac{1}{|\mathfrak{G}|-1}\sum_{i=1}^{|\mathfrak{G}|-1} \mathcal{S}_1^*(\mathfrak{C}_{i+1}), \tag{4}$$

$$\mathcal{H}(\mathfrak{G}) = \frac{1}{|\mathfrak{G}|-1}\sum_{i=1}^{|\mathfrak{G}|-1} \mathcal{H}_1^*(\mathfrak{C}_{i+1}), \tag{5}$$

whereas fitness function f_c^* for a clustering $\mathfrak{C}$ has been defined as a maximization problem in the way:

$$f_c^*(\mathfrak{C}) = \langle \mathcal{S}_1^*(\mathfrak{C}), \max\mathfrak{D} - \mathcal{H}_1^*(\mathfrak{C})\rangle. \tag{6}$$

Once the process to measure the clustering and dendrogram fitness, has been completed, we must define the selection and solution comparison process since now the previous fitness functions have not been defined to give a single fitness value. The selection method is *tournament selection* (Goldberg, 1989), in which the genetic algorithm first selects all the non-dominated individuals (*the Pareto front*) of the current population to be part of the mating process. The remaining individuals in the population are then chosen by using tournament selection. In addition, an elitism (the most fit individual) has been passed from generation to generation. The solution comparison process is focused on the idea given in Pappa et al. (2002), which consists of involving a total order (in mathematical terms) able to compare those non-dominated (non-comparable) individuals, since the dominance concept imposes a partial order on the individuals, i.e., it is not always possible to decide which individual is better.

Thus, the following tie-breaking criterion has been proposed by following the principle of Pareto dominance: given two non-dominated individuals (dendrograms) Id_1 and Id_2, we compute the number of individuals dominated in the current population by Id_1 as $d_{1>}$ and the number of individuals dominating Id_1 as $d_{1<}$. The same process is carried out for Id_2 to obtain $d_{2>}$ and $d_{2<}$. Then, the individual reaching the highest score from set $\{d_{1>} - d_{1<}, d_{2>} - d_{2<}\}$ is the winner. If the differences above have the same score, the winner individual is then selected randomly. This way, the comparison process of individuals is completed.

3.6.2. Genetic operators

The two genetic operators used in ECM follow the strategy given in Castellanos-Garzón and Díaz (2013), which include details of both genetic operators and the ECM search method, however a little explanation of the operators will be given here. The mutation operator (MO) is a unitary alteration which is applied to a single dendrogram by exploring its different branches. Hence, the MO carries out an in-depth search. Only a part of the transformed dendrogram is modified with this operator and the other is kept unchangeable. Indeed, since a dendrogram is a special kind of tree, this MO works similar to moving a cluster associated with a branch of the dendrogram to another branch in the same dendrogram.

The crossover operator (CO) recombines valuable information from two individuals in order to yield a single individual, which inherits the genetic code of their ancestors. Thus, this operator is responsible for carrying out a wide search in the dendrogram space. In general terms, the CO randomly chooses the same level from two parent dendrograms to form a new clustering (which is called seed clustering) by selecting the best clusters from parents in the chosen level, i.e., for each clustering, the half of the best clusters is selected to form the seed clustering. After that, the child dendrogram is built by applying the MO on the seed clustering to achieve the upper levels. Finally, a divisive strategy (for clusters) is also applied on the seed clustering to build the remaining lower levels.

4. Case study on pancreatic ductal adenocarcinoma

This section describes the case study used in this research as well as the results of the proposed framework applied to it. We outline the main characteristics of the target dataset, Pancreatic Ductal Adenocarcinoma (PDAC) and the specific setting imposed on framework HybridFrame (as its modules and methods) to achieve informative gene sets from dataset PDAC. In this context, we also introduce two comparison processes of the results given by our framework with respect to the boundary point contribution (with and without boundary genes) and other gene selection methods. Furthermore, at the end of this section we discuss the results obtained, which have also been supported through HybridFrame visualizations given in Appendix. The experiments have been carried out in a computer with a RAM = 8 GB, CPU = $InterCorei$5-44603.20 GHz and *Operating System* = *Windows 8.1 Pro (64 bits)*.

4.1. PDAC dataset

As explained in the Introduction, the goal pursued by our proposal on PDAC is to discover possible informative gene subsets (as small as possible), allowing a further pharmaceutical research of some of those informative genes, because the drug's low effectiveness in the PDAC treatment is well known, Liss and Thayer (2012). The specific used PDAC dataset comes from The National Center for Biotechnology Information (NCBI), http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE15471 public repository. The study has been provided by Badea et al. (2008a, b) and focused on an expression analysis of 36 pancreatic ductal adenocarcinoma tumors and on matching normal pancreatic tissue samples from pancreatic cancer patients of the Clinical Institute Fundeni (ICF) using *Affymetrix U133 Plus 2.0 whole-genome microarray chips*. Pairs of normal and tumor tissue samples were obtained at the time of surgery from the resected pancreas of 36 pancreatic cancer patients. As a final result of this study, a PDAC dataset (gene expression matrix) with 54 675 gene-probes against 78 patient tissue samples was achieved and normalized using the RMA algorithm (Robust Multichip Average).

4.2. Running module SFM on dataset PDAC

For this case study, SFM runs three processes in a chained way from PDAC given as its input, namely: data processing, the Mann–Whitney and Kruskal–Wallis test. In the data processing, data have been standardized to mean 0 and variance 1. Next, the Mann–Whitney and Kruskal–Wallis tests are applied as chined filter methods. Mann–Whitney is used to filter genes whose p-value < 0.05, i.e., genes whose variation of their gene expression level is explained by means of relation normal/tumor tissue given in dataset PDAC (onwards, PDAC for short). Once this test was applied to PDAC, 31 850 gene-probes were filtered out to a new dataset as the most meaningful probes related to the disease. After the step above, SFM applies the Kruskal–Wallis test to resulting data in order to extract those probes more closely related to age of patients (as a correlation analysis), since age appears to be a factor influencing the severity of the disease. To do this, the age variable representing age of patients from 45 to 77 years old was introduced. But this test requires to state age subgroups (intervals), which have been given as listed in Table 1. Then, the previous 31 850 gene-probes were analyzed with the Kruskal–Wallis test to filter out probes with p-value < 0.05, which achieved a dataset with 1299 probes. It should be noted that the probes above are those most closely related to the disease and age of patients. Figs. 2 and 3 show an overview of the stages involved in the PDAC filtering process (on x-axis) against the number of probes filtered out (on y-axis) for methods CIM and ECM respectively, given by HybridFrame.

Table 1
Age subgroups established for correlation analysis of age with gene expression level.

Meaning	Group#1	Group#2	Group#3	Group#4
Age interval	[45, 54]	[55, 61]	[63, 67]	[68, 77]
Patient number	9	9	8	10

4.3. Running module HCMM

The next module to run is HCMM (see Fig. 1). The idea followed by the framework in the module above SFM is to reduce noise of the dataset as much as possible before applying any HCMM clustering method. This ensures a better performance on the applied clustering methods since the clustering process will not be affected by the use of irrelevant genes (acting as noise) for the studied disease. Therefore, the dataset with 1299 gene-probes given as input to HCMM contains the most interesting genes to be analyzed in subsequent phases. The task of HCMM is to then cluster the data by using different strategies to process and recombine different cluster results in the later stages. To do this, the five hierarchical clustering methods prefixed in HCMM were selected, i.e., Agnes, Diana, Eisen, HybridHClust and TSVQ. The Euclidean distance was chosen for all methods and *average* was chosen as inter-cluster distance for the Agnes and Diana methods. Finally, five dendrograms were given as output after applying the selected methods to the input dataset (1299 gene-probes).

4.4. Running module VAM

The HCMM output goes to both the VAM module and ECM method but before dealing with VAM, we will first give the setting for method ECM since its output (Part-(A)) also goes to the input of VAM. The ECM initial population consists of the five dendrograms given from different clustering methods of HCMM and its setting to evolve those individuals was listed in Table 2. Parameters δ, τ, ϵ and α given in this table are internal parameters prefixed in ECM, which can be consulted in Castellanos-Garzón and Díaz (2013).

Meanwhile, the VAM module starts to process the output of HCMM and ECM. To do so, it first estimates the dendrogram level intervals with the best clusterings based on internal measures of cluster validation; that is, *separation, homogeneity* and *silhouette width*. From the values given by these three measures for each clustering of each dendrogram, we have created level intervals for each dendrogram based on the two clustering levels reaching the maximum and minimum score. That is, for each dendrogram, the levels reaching the best scores for each measure are selected and then, the interval to be chosen is formed by the maximum and minimum level number from the previously selected levels by each measure. This way, we would have certain assurance that clusterings in that interval meet more than one cluster validation criterion, which is desirable to guide the user in the process of clustering selection from a given dendrogram. Keep in mind that in the chosen interval are the three clusterings whose used measure scores were the highest. Moreover, since clusterings in a dendrogram are nested, we have that their scores in the

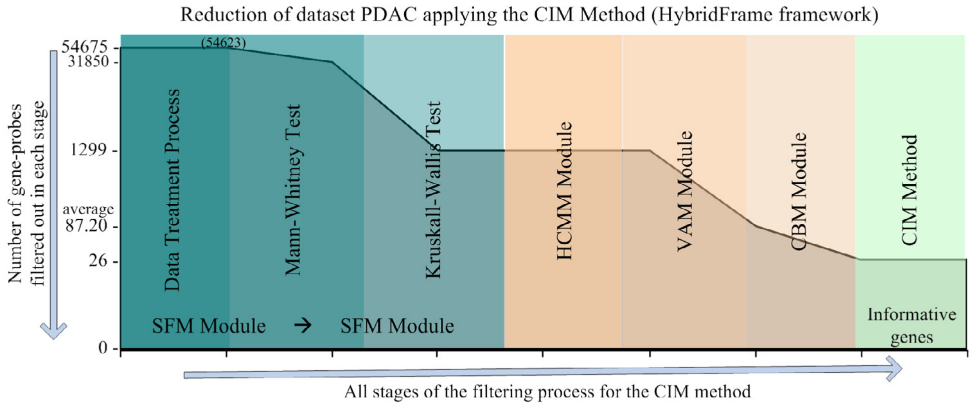

Fig. 2. Chart summarizing the filtering stages used by the CIM method in HybridFrame to reduce PDAC until reaching an informative gene subset. The diagram shows the involved filter processes vs. the number of probes filtered out after running each stage. The shaded area under the curve represents the remaining probe portion of PDAC when the filtering process is applied in 7 stages. At the end of the process, a subset of 26 informative genes is reached.

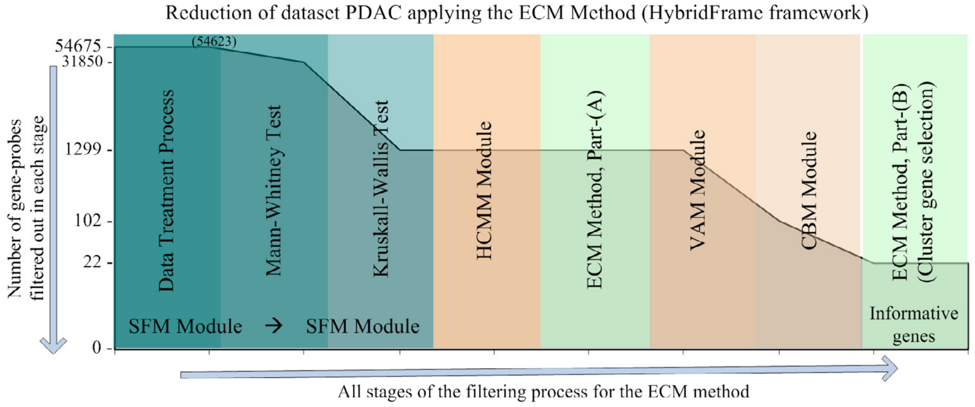

Fig. 3. Chart summarizing the filtering stages used by the ECM method in HybridFrame to reduce PDAC until reaching an informative gene subset. The diagram shows the involved filter processes vs. the number of probes filtered out after running each stage. The shaded area under the curve represents the remaining probe portion of PDAC when the filtering process has been applied in 8 stages. At the end of the process, a subset of 22 informative genes is reached.

Table 2
Parameter settings to run evolutionary method ECM on PDAC.

Parameter	Value (or interval)
Crossover probability	[0.60, 0.75]
Mutation probability	[0.10, 0.20]
Number of individuals	30
Number of generations	$[10^3, 10^6]$
δ	15/16
τ	[0.15, 0.40]
ϵ	0.03
α	0.90

Table 3
Level intervals selected for each dendrogram of the clustering methods applied to PDAC. The level finally selected (clustering) by the visual validation process as well as its corresponding number of clusters are also listed.

Method	Level interval	Chosen level	Cluster number
Agnes	[1288, 1298]	1293	7
Diana	[1294, 1298]	1287	13
Eisen	[1297, 1298]	1287	13
HybridHClust	[1291, 1298]	1292	8
TSVQ	[1291, 1298]	1292	8
ECM	[1289, 1298]	1290	10

selected interval will not be very different from the three clusterings with maximum scores, in most cases.

As a result of the process above, Table 3 lists in column *Level interval*, the level intervals selected for each dendrogram whereas columns *Selected level* and *Cluster number* show the selected level and its number of clusters, respectively, given by the visualization process (Visual Cluster Validation) of VAM, which is the second process to be applied.

4.4.1. *Visual cluster validation*

Before going on to the VAM visualization process, note that the intervals estimated in Table 3 only provide a guideline of where to start exploring dendrograms in the visualization process looking for a suitable clustering; this means that the final selected clustering does not necessarily have to be in the given interval, as in the case of the Diana and Eisen methods shown in this table. Then, the strategy of process Visual Cluster Validation is to find a suitable clustering by comparing different cluster visualizations taking into account the score reached by the applied validity measures, thus validating the choice made by the user.

Then, the visualization sequences followed by VAM to select a clustering are shown in Fig. A.1 of Appendix, which also reinforces the results given in Table 3. VAM brings linked views of dendrogram, heatmap, parallel coordinates and genes (as well as boundary genes) displayed as 3D-points on a scatterplot. All aim to guide the user in the selection process. Completing this module, Fig. A.2 in Appendix shows the clusterings finally selected by VAM in the form of dendrograms on microarray-heatmaps. Note that this figure represents and supports the results given in columns 3 and 4 of Table 3.

4.5. Running module CBM

CBM is the last module applied to the PDAC filtering process and computes the boundary genes of the clusterings selected in the VAM module. Thus, new clusterings are created from the previous ones by finding genes in the boundary of each cluster coming from the input clusterings to CBM. Hence, a clustering boundary has been achieved from the result of method ECM (Part-(A)) and five clustering boundaries have also been obtained for method CIM. Then, an output goes to Part-(B) of method ECM to finally select informative genes according to this method. The other outputs of CBM go to method CIM, which establishes intersection levels to compute informative genes.

4.5.1. Running method ECM

The strategy followed by ECM to select informative genes (see options given in Part-(B) of ECM, Fig. 1) from each boundary cluster is to combine significance (p-value), variance and graphics of parallel coordinates on genes of each cluster. Then, we have defined an objective function to select 25% of the genes in each cluster (with the highest scores) while such a gene selection is being also supported by parallel coordinate exploration. The objective function is defined as follows:

$$Score(g) := \alpha_1 \cdot significance(g) + \alpha_2 \cdot variance(g), \tag{7}$$

where g is a gene and α_1, α_2 are scalars which can be defined as $\alpha_1 = -1$ since the gene significance is usually in real interval $[0, 1]$ and $\alpha_2 = \frac{1}{maxvar}$. $maxvar$ is the maximum gene variance computed from the dataset. As a result, the larger the values of function Score the higher the gene relevance, which requires finding small values for $significance$ against big values for $variance$ as a maximization process. Parallel coordinate graphics showing genes of each boundary cluster from method ECM are displayed in Fig. A.3, Appendix. Curves represent genes displayed from patient age against gene expression level. The application of the objective function above in combination with Fig. A.3 from Appendix resulted in a subset of 22 informative genes (see full process in Fig. 3). We want to stress that when we talk about combination of the Score function with a parallel coordinate graphic (PCG), we are referring to the genes selected by Score are validated through the PCG. Thus, genes showing similar profiles (similar curves) in the PCG are removed, i.e., only one gene (reaching the highest score) is selected from such genes. Conversely, if a gene showing a different profile from the rest in the PCG has not been selected by Score, then that gene is added to the set of informative genes.

Reinforcing the result given by ECM, Fig. A.5 in Appendix shows another parallel coordinate graphic visualizing the 22 informative genes in form of curves as presented in Fig. A.3. Note that in general, there exists a low correlation between the genes displayed in this figure.

4.5.2. Running method CIM

As previously explained, the remaining clustering boundary outputs of CBM coming from different clustering methods are processed by the CIM filter method. In this case, a result of 26 boundary genes was reached in the first intersection level of the algorithm (see Fig. 2 for the whole process), which implies that such genes have the highest significance and, consequently, they form an informative gene set. Supporting this result for CIM on PDAC, Fig. A.4 in Appendix displays a parallel coordinate graphic representing each gene of the result. Curves in those graphics represent genes displayed from the patient's age against gene expression level. Note that as with Fig. A.5, genes involved in Fig. A.4 show a low correlation between them. Moreover, both parallel coordinate graphics (Figs. A.4 and A.5 in Appendix) provide a means for a further reduction of both found informative gene subsets, if needed. The latter will be seen in Section 4.6.2.

4.6. Results

This subsection details the final results of HybridFrame given through its two filter methods ECM and CIM. For this propose, we have outlined two tables, one for each filter method, identifying informative genes of each method. The tables show information such as, gene identifier, gene name and whether such a gene has previously been identified in other research and/or databases as a PDAC-related gene. Information provided by both tables was consulted in PED (http://www.pancreasexpression.org/) and Pancreatic Cancer Database (http://pancreaticcancerdatabase.org/index.php). Tables 4 and 5 list informative genes from PDAC by using the ECM and CIM methods respectively. Those genes have a larger relation to normal and tumor tissue samples of PDAC and are highly age-related. Moreover, 10 genes from these tables were identified by both methods (genes in the intersection are highlighted in both tables), meaning they could be even more meaningful for PDAC than the rest. In summary, according to the whole discovery process of informative genes given by Hybridframe, we assume that genes in Tables 4 and 5 can be considered for further pharmaceutic research.

4.6.1. Assessing module CBM of HybridFrame

This subsection is in charge of assessing the importance of the CBM module (boundary gene module) in the HybridFrame framework. Despite the fact that this case study has been oriented to biomarker discovery by showing the relevance of the found genes in relation to the age of patients and from the biological point of view, i.e., in a qualitative way; we want to also assess the impact of those genes from the quantitative point of view. In this case, a measure evaluating the accuracy of such genes in classification tasks will be given.

Thus, the aim of this subsection is to measure the contribution of the CBM module to HybridFrame, i.e, the importance of the boundary genes in the gene selection process. Hence, the accuracy of the genes found by HybridFrame with and without the CBM module will be measured. To do this, a classifier based on k-nearest neighbors (kNN) will be used, Tan et al. (2006). kNN has been selected because it is one of the simplest but effective classification models and in addition, it develops a lazy model, which does not need to rebuild the learning model against changes in the training set, as with other classifiers. Then, to run HybridFrame without the CBM module, the output of the VAM module is connected to the inputs of both Part-(B) (see Fig. 1) of the ECM method and CIM method. Note that for the case of the CIM method, a gene selection process must be applied to each cluster of each input clustering before running CIM, since boundary genes are not computed. In this case, the Score function given in (7) is applied to each cluster of each clustering by taking out 12% of genes in each cluster. The results from the input clusterings are intersected through the CIM method. As for Part-(B) of the ECM method, we have that the same strategy as in Section 4.5.1 is applied to its input clustering, but in this case, the gene percentage selected in each cluster is 12%.

Note that in this case, a gene percentage smaller than the one for boundary clusters has been chosen from each cluster. The reason is that we are interested in achieving small subsets of informative genes. Since a cluster has much more genes than a boundary cluster, then we must filter a number of genes smaller than 25%. Finally, once the strategy to run HybridFrame without the CBM module has been defined, we have that methods ECM and CIM obtained 162 and 128 genes respectively. Table 6 shows an accuracy comparative with the kNN classifier on the ECM and CIM results with/without boundary genes for HybridFrame applied to PDAC. The accuracy for each case has been computed by using methodology stratified tenfold cross-validation, Flach (2012). The table structure is as follows: column *Method* lists the name of the method applied in each case, *Number of genes* is the number of genes discovered by each method, K is the number of neighbors used by kNN in the classification process whereas *kNN-accuracy* is the accuracy percentage reached by each method applied.

Table 4
26 informative genes given from PDAC by the ECM filter method in HybridFrame. The gene identifier, name and previous identification are listed. Genes also discovered by the CIM method; that is, genes in the intersection of both methods have been highlighted.

Identifier	Gene name	Previously identified
C3	complement component 3	Yes
COL6A3	collagen type VI alpha 3	Yes
FBN1	fibrillin 1	Yes
FSTL1	follistatin-like 1	Yes
DPYSL3	dihydropyrimidinase like 3	Yes
TNFAIP3	TNF alpha induced protein 3	Yes
NPIPB5	nuclear pore complex interacting protein family, member B5	No
FERMT2	fermitin family member 2	Yes
TCF4	transcription factor 4	Yes
CLDN11	claudin 11	Yes
C1QTNF3	C1q and tumor necrosis factor related protein 3	Yes
SPON1	spondin 1	Yes
NRK	Nik related kinase	Yes
GZMB	granzyme B	Yes
PTPRC	protein tyrosine phosphatase, receptor type C	Yes
DEFA6	defensin alpha 6	Yes[a]
FAM198A	family with sequence similarity 198 member A	No
ISLR	immunoglobulin superfamily containing leucine-rich repeat	Yes
TNFRSF12A	tumor necrosis factor receptor superfamily member 12A	Yes
CXCL5	chemokine (C-X-C motif) ligand 5	Yes
CCL25	chemokine (C–C motif) ligand 25	Yes
CLEC4M	C-type lectin domain family 4 member M	Yes[a]
KRT13	keratin 13	Yes
SEL1L	SEL1L ERAD E3 ligase adaptor subunit	Yes
HMCN1	hemicentin 1	Yes
TSHZ2	teashirt zinc finger homeobox 2	Yes

[a] Genes previously identified in pancreatic cancer without any specified subtype.

Table 5
22 informative genes given from PDAC by the CIM filter method in HybridFrame. The gene identifier, name and previous identification are listed. Genes also discovered by the ECM method; that is, genes in the intersection of both methods have been highlighted.

Identifier	Gene name	Previously identified
NKIRAS1	NFKB inhibitor interacting Ras-like 1	No
TNFAIP3	TNF alpha induced protein 3	Yes
BICC1	BicC family RNA binding protein 1	Yes
SPON1	spondin 1	Yes
ENTPD1	ectonucleoside triphosphate diphosphohydrolase 1	Yes
CXCL5	chemokine (C-X-C motif) ligand 5	Yes
GZMB	granzyme B	Yes
PEG3	paternally expressed 3	Yes
PTPRC	protein tyrosine phosphatase, receptor type C	Yes
KRT13	keratin 13	Yes
SEL1L	SEL1L ERAD E3 ligase adaptor subunit	Yes
COPZ1	coatomer protein complex subunit zeta 1	Yes[a]
CLEC4M	C-type lectin domain family 4 member M	Yes[a]
C3	complement component 3	Yes
NRK	Nik related kinase	Yes
AFAP1-AS1	AFAP1 antisense RNA 1	No
GSTO2	glutathione S-transferase omega 2	Yes
GBA3	glucosidase, beta, acid 3	Yes[a]
PRSS35	protease, serine 35	Yes[b]
SAMSN1	SAM domain, SH3 domain and nuclear localization signals 1	Yes
MYLK3	myosin light chain kinase 3	No
RPL37A	ribosomal protein L37a	Yes

[a] Genes identified in pancreatic cancer without any specified subtype.

[b] Genes identified in other subtypes of pancreatic cancer.

Table 6
Result comparative table of gene selection methods from framework HybridFrame with/without module CBM for dataset PDAC. The results have been evaluated through the accuracy measure by means of classifier kNN.

Method	Number of genes	K	kNN-accuracy (%)
Method ECM without module CBM	162	3	84.72
Method ECM with module CBM	26	6	90.28
Method CIM without module CBM	128	3	86.11
Method CIM with module CBM	22	4	90.28

Table 7
Comparative table of gene selection methods applied to dataset PDAC. Methods ECM and CIM of framework HybridFrame have been compared with six gene selection methods. The number of genes, its accuracy by using a kNN classifier and the runtime taken by each method have been listed.

Method	Number of genes	K	kNN-accuracy (%)	Runtime
propOverlap	1123	5	86.11	0.11 min
Boruta	10	1	91.67	26.63 min
SDA	5	7	88.89	0.10 min
Spikeslab	37	3	90.28	4.86 min
kofnGA	5	1	72.22	14.91 h
FSM	25	1	88.89	[5.04, 10.04] h
HybridFrame:				
ECM	2	5	91.67	[3.27, 6.27] h
CIM	3	22	91.67	11.52 min

As shown in this table, the HybridFrame version with the CBM module has reached the best accuracy across the kNN classifier. Hence, this proves that the module of boundary genes is significant for the framework. On the other hand, note that the boundary point algorithm reduces much more the number of genes selected for each cluster than the HybridFrame version without the CBM module.

4.6.2. *Comparing HybridFrame with respect to other methods*

As mentioned in the subsection above, although the aim of this research has been biomarker discovery evaluated from the biological point of view, we have compared the found gene subsets with the results of other recent gene selection methods. This will allow us to evaluate and compare our proposal not only for biological purposes, but also in a quantitative way through classification tasks with a classifier kNN. The comparison has been made with respect to the following gene selection methods: *propOverlap* in Mahmoud et al. (2014, 2015), *Boruta* in Kursa and Rudnicki (2010, 2016), *kofnGA* in Wolters (2015b, a), *SDA* in Ahdesmaki and Strimmer (2010); Ahdesmaki et al. (2015), *Spikeslab* in Ishwaran and Rao (2005); Ishwaran et al. (2013) and *FSM* in Castellanos-Garzón et al. (2016). The parameters of these methods have been configured according to their default values (as defined by each method), except for kofnGA which is a genetic algorithm, whose parameters not assigned by default were set as follows: size of initial population to 100 and number of generations to 9000.

Table 7 lists a comparative of accuracy, number of genes and runtime for each of methods above with respect to our proposal, HybridFrame (ECM and CIM methods). The runtime (or runtime interval) taken by each method has been given in minutes (mins) or hours as applicable. The accuracy reached for each method has been computed as a stratified tenfold cross-validation and the table structure is the same as the one of Table 6. Moreover, in order to make the HybridFrame results more competitive in classification tasks when HybridFrame is compared with other methods, the number of genes listed in Tables 4 (26 genes) and 5 (22 genes) from methods ECM and CIM respectively, has been reduced a gene minimum (2 and 3 genes as shown in Table 7), maximizing their accuracies. To find the genes above from Tables 4 and 5, we have used the objective function given in (7) in combination with parallel coordinate graphics as done in Section 4.5.1.

As shown in this table, the methods reaching the best results have been stressed along with their accuracies. The smallest gene subset has also been underlined. Methods ECM, CIM and Boruta have achieved the best accuracy (91.67%) whereas method ECM has also reached the smallest number of genes (2 genes). This way, the genes found by our proposal have been 2 genes for ECM which are: {*COL6A3, ISLR*} and 3 genes for CIM which are: {*SPON1, CXCL5, C3*}. Finally, note that both methods of HybridFrame have held the main goals expected in the gene selection process for biomarker discovery and disease classification, which are a small number of genes and that such genes disclose high accuracy.

4.6.3. *Discussion*

This subsection provides a discussion on the methodology used and the final results given in Tables 4–7 for PDAC. The proposed framework has discovered two small sets of genes altered in PDAC, i.e., differentially expressed genes. Those genes are related statistically to the study factor and have allowed us to evaluate the impact of age in the transcriptome of PDAC tissue samples. To reach the results above, an analysis of biological consistency of the discovered genes was carried out according to their involvement in different cellular processes. The results of the analysis indicate that such genes are highly related to pancreatic cancer and some present a direct involvement. Another important aspect of this analysis is that the proposed framework has been able to identify previously unidentified genes as PDAC-related. This fact suggests further research to gain insight into the involvement of such genes in PDAC.

Although we are not going to explain the functions performed by each one of the discovered genes, those functions were studied to support the validation process of gene selection. For example, the first gene given in Table 5, gene NKIRAS1, which has not previously been identified according to literature can be important from a functional point of view. This gene is involved in one of the main cell growth and embryogenic development pathways (NF-kappa B), commonly associated with cancer. In this pathway, the NKIRAS1 protein prevents the degradation of NF-kappa B inhibitor beta (NFKBIB) acting as a regulator of NF-kappa B activity (Uniprot, http://www.uniprot.org/). Therefore, an altered expression in this gene may have implications in cell growth. Another example to consider is the SPON1 gene given in Tables 4, 5 and method CIM in the section above, which is one of the 10 most significant genes selected by the two HybridFrame methods. Moreover, different probes of this gene have appeared in the final results. Therefore, *SPON1* appears to be the most significant gene. Additionally, it encodes an extracellular matrix protein contributing to the growth of axons in spinal cord. It should be noted that genes encoding matrix proteins are commonly altered in PDAC, Liss and Thayer (2012).

For its part, Tables 6 and 7 also support the reliability of the HybridFrame framework in different areas such as, biomarker discovery and disease machine learning. On the one hand, Table 6 shows that the CBM module actually improves the results of the framework and on the other hand, Table 7 shows that HybridFrame can reach better results than the existing methods, which completes its importance as a filter method. However, as shown in Table 7, the HybridFrame runtime is greater than the runtimes of the remaining methods (except for the Spikeslab and FSM methods), meanly when the ECM method of HybridFrame is run. The ECM evolutionary nature makes it depend on the runtime assigned by the user for its convergence. In this case, the runtime assigned to ECM has been between 2 and 5 h approximately. The above increases the overall runtime of HybridFrame, although this fact is justified by the achieved results and when HybridFrame is compared with other methods in classification tasks. In this sense, we want to stress that even though one of the methods in Table 7 (Boruta method) achieved the same accuracy as that of HybridFrame, our framework reached a smaller number of genes. Hence, our framework provides a high filtering capacity, allowing us to obtain small sets of biomarkers for classification purposes, which is an important advantage in diagnosis applications.

Since HybridFrame is considered a composite method, it develops a set of hybrid techniques from data mining to achieve a more complex filtering process than a simple method. In particular, the introduction of cluster boundary points to the gene filtering process of HybridFrame has been key in the discovery of informative genes. HybridFrame provides a modular and flexible structure, allowing us to add new components, besides that it can be applied to different studies of gene expression data. HybridFrame also provides two filter methods, ECM and CIM. Both methods have performed well, with the ability to find very promising solutions from a biological point of view. On the one hand, ECM can find better solutions than CIM since ECM uses the evolutionary force

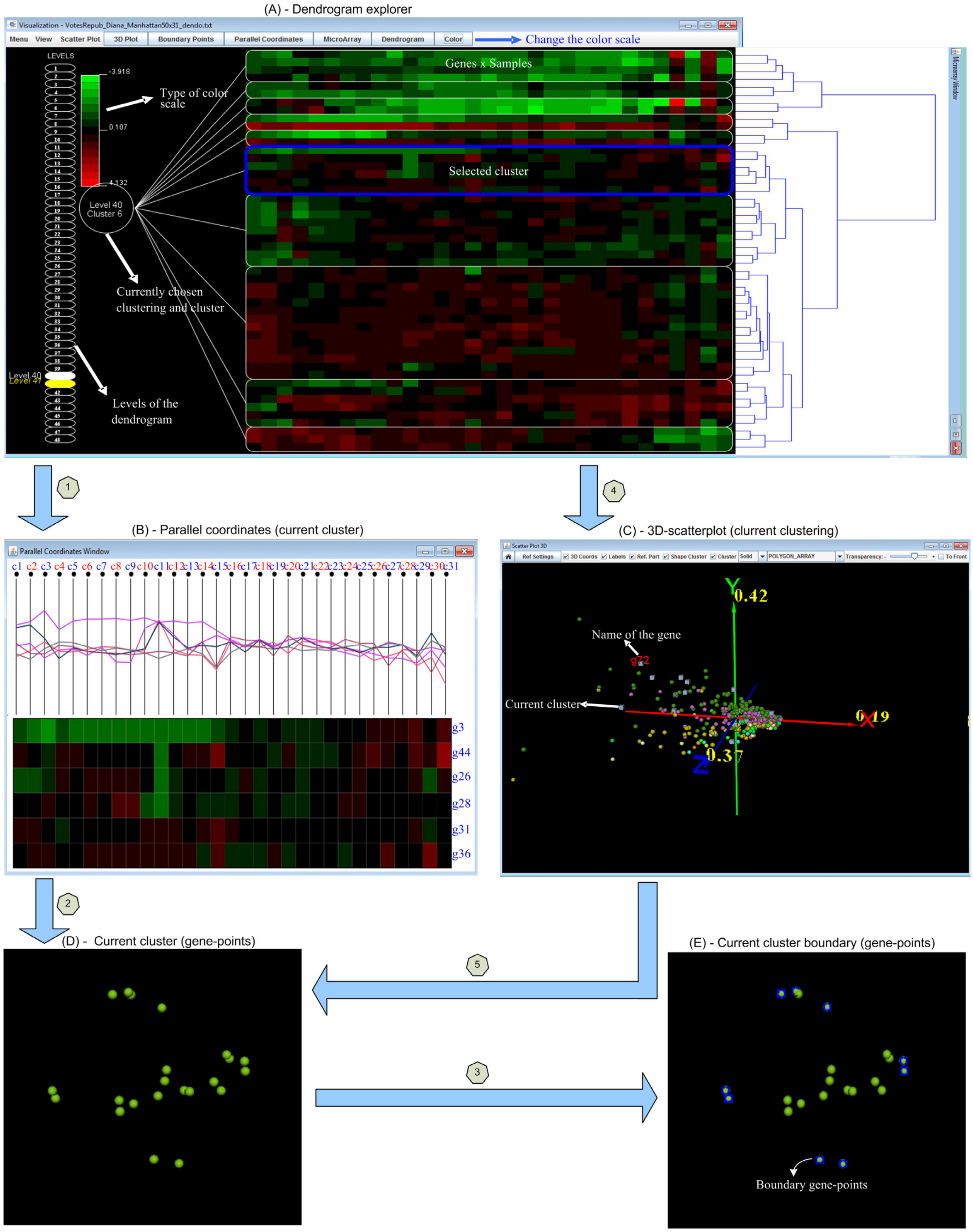

Fig. A.1. Workflow representing the visualization sequences followed to select the most suitable clustering from a dendrogram. There are two visualization sequences to follow, ranging from dendrogram global views (View-(A)) to views focusing on details (by a zoom-in) of the selected level and/or cluster. The first view sequence to validate a selected level is $\langle (A) - (B) - (D) - (E) \rangle$ and the second is $\langle (A) - (C) - (D) - (E) \rangle$.

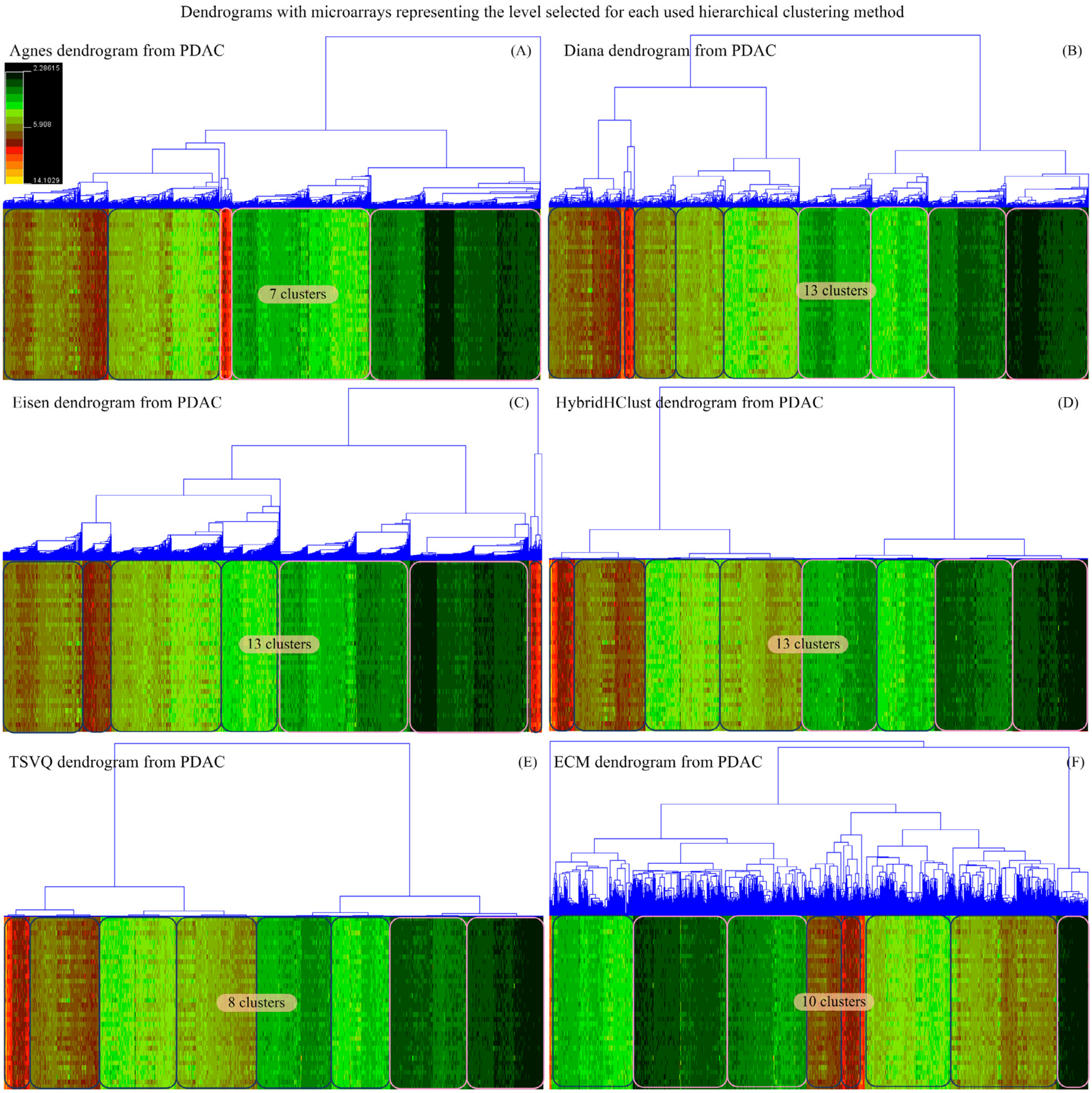

Fig. A.2. Clusterings selected from each dendrogram given by the hierarchical clustering methods used on PDAC. The clusterings are selected by means of the processes involved in module VAM of HybridFrame. Clustering methods run on a subset of 1299 genes from PDAC.

to improve solutions given by other methods. However, it includes a gene selection process from clusters in which techniques selected for this propose are chosen by the user. Therefore, special care must be taken when selecting such techniques because the results could be affected drastically. On the other hand, CIM is a fully automatic (no user intervention required) and very fast method compared to ECM. Moreover, the fact that the user does not intervene in the process removes the possibility of introducing bias to the solutions.

To conclude on this section, we have that one of the goals of this research has been the study of a possible influence of the age factor on gene expression levels from PDAC. The results of our study indicate that despite the fact that the parallel coordinate graphics given in Figs. A.3–A.5 present a slight decrease in the gene expression level with the increase of age, such a fact cannot yet be decisively claimed for a general result in PDAC. We can say, however, that according to this study, age is not a determining factor for the expression levels of the selected genes once the disease has developed. Therefore, there is no difference with respect to the patient's prognosis when the age factor is involved. Note that the above does not mean that age is not a risk factor in the development of this type of cancer, since it is well-known that carcinogenic processes and age have an undeniable relation as the aging process causes a gradual accumulation of cellular damage, Nicolai et al. (2015).

5. Conclusions

This paper has proposed a data mining framework aimed at the gene selection process from DNA-microarray data. In this context, the framework has developed a strategy to successively reduce an input dataset, until reaching a set of informative genes. To achieve this, the framework was based on both statistical and data mining techniques to create a hybrid technique environment, which enabled the framework

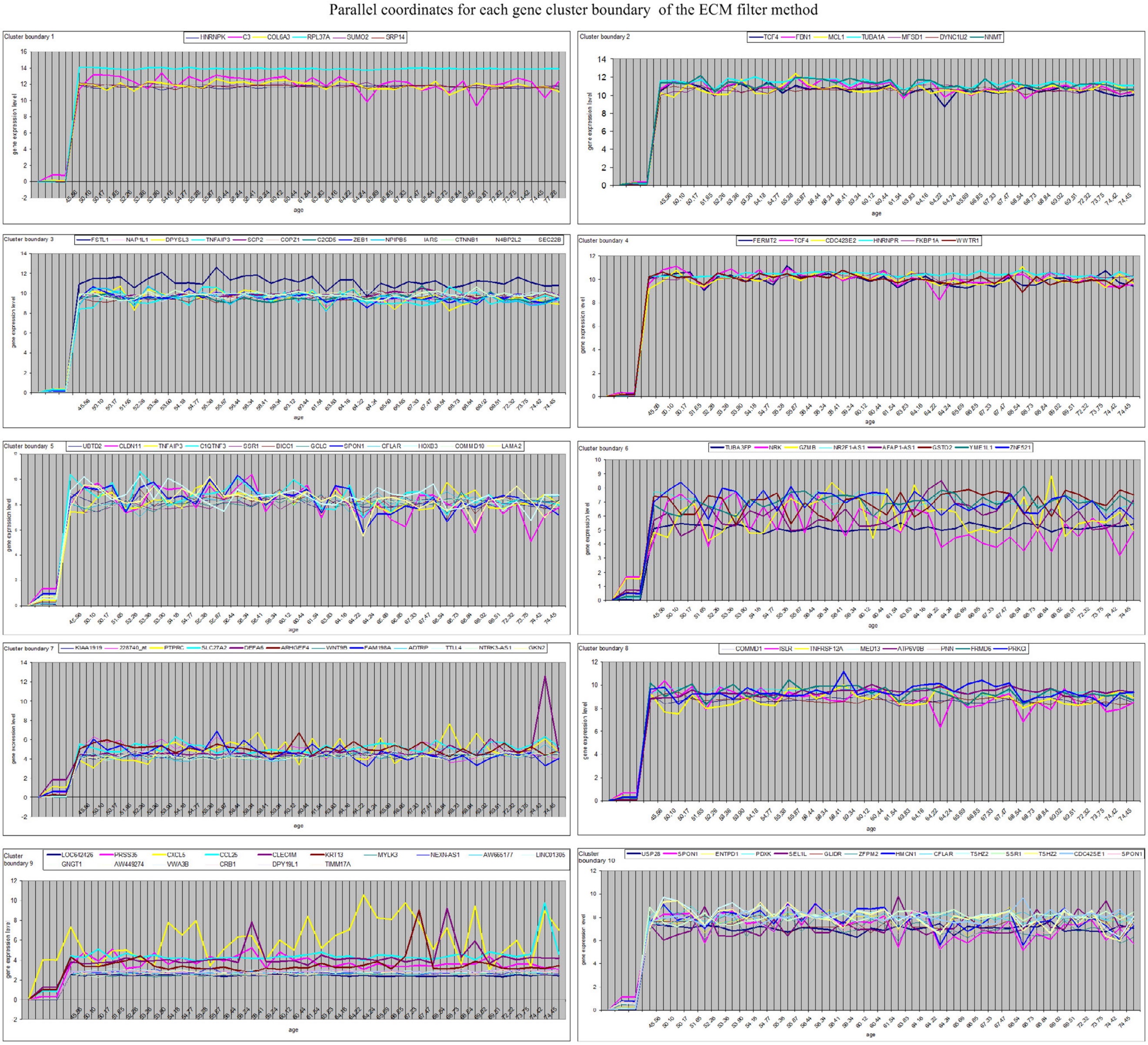

Fig. A.3. 10 parallel coordinate charts associated with each cluster boundary of the ECM filter method. Curves in each chart represent genes in each boundary, inspecting patient age vs. gene expression level.

to define two different gene filter methods. The methodology followed by the framework has proven to be effective in the gene selection process, offering a consistent selection from a biological point of view and classification tasks. The framework was designed in a flexible way, allowing us to add new filter and clustering methods to the process. It has also allowed us to validate the given gene clusters as well as the final results through linked cluster visualizations. Moreover, the gene subsets from PDAC, discovered by the filter methods given by the framework provide a starting point for laboratory researchers. Hence, our methodology can contribute to gaining insight into molecular processes of cancer by facing different aims such as biomarker research, pharmaceutic applications and the influence study of different factors in gene expression levels.

The result of the analysis carried out on the PDAC case study, indicate that the applied methodology has not only been able to find previously identified genes, it has also been able to discover still unidentified genes, suggesting further research to determine their relationship to the kind of cancer. In this sense, we have also studied whether there is a significant influence of age on gene expression levels given in PDAC patients. Although data have disclosed a slight tendency to decrease gene expression levels when age increases, the age factor has not been found to be determining in the expression changes of the selected genes when the pathology has already developed. Finally, we stress the fact that a key point in the development of this research has been the application of a boundary point approach to the gene selection process, which is a novelty in this field. Therefore, all previous contributions and results prove that our approach can be very useful in the analysis and knowledge discovery process from DNA-microarray data.

Acknowledgment

We would like to thank Dr. Liviu Badea from Bioinformatics research group, National Institute for Research in Informatics (Rumania) for provided additional information on the dataset used in this research. The research of Juan Ramos González has been co-financed by the European Social Fund and Junta de Castilla y León (Operational Programme 2014-2020 for Castilla y León, BOCYL EDU/602/2016). This work has also been supported by project MOVIURBAN: Máquina social para la gestión

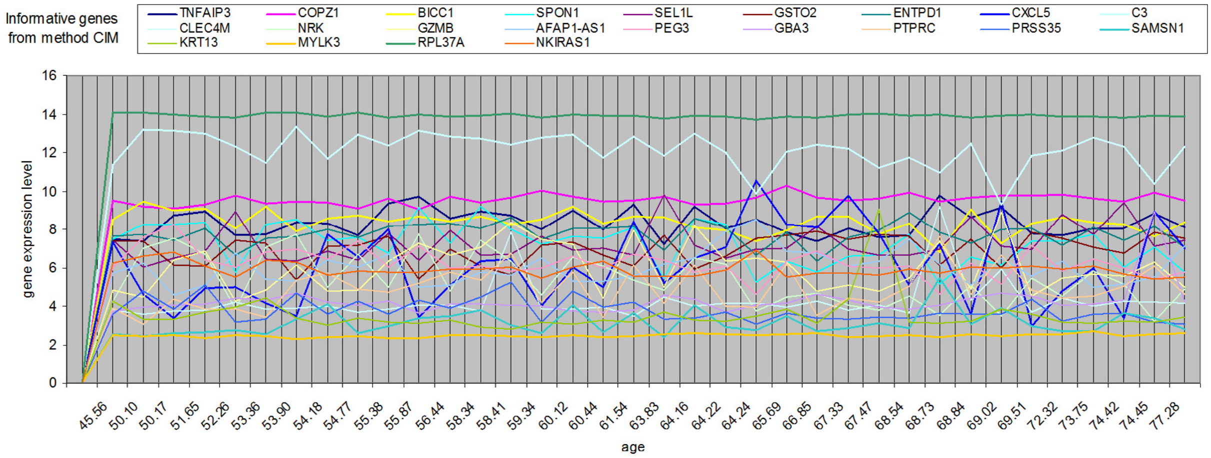

Fig. A.4. Parallel coordinate chart displaying each informative gene (22 genes) from PDAC given by the CIM filter method. Curves represent genes evaluated by patient age against gene expression level.

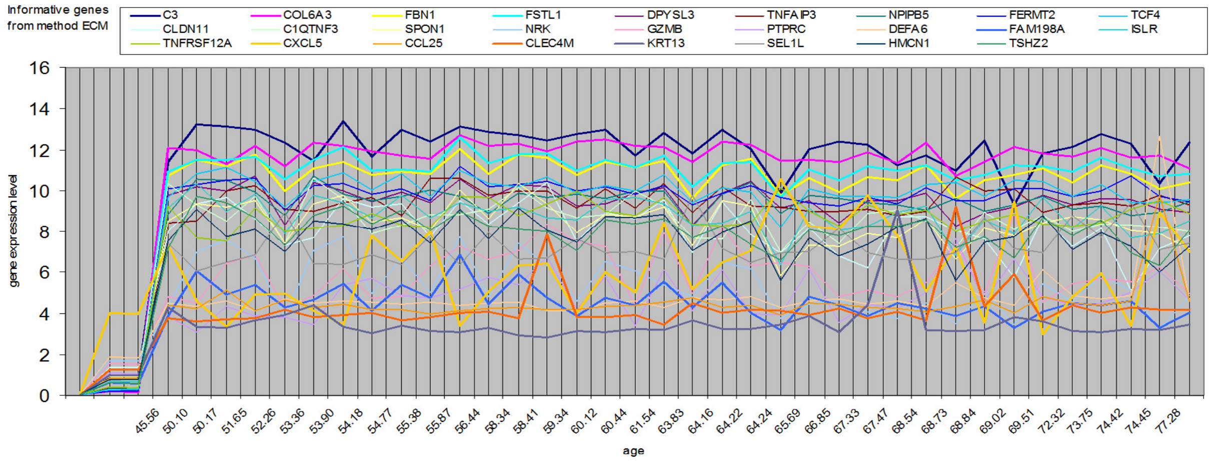

Fig. A.5. Parallel coordinate chart displaying each informative gene (26 genes) from PDAC given by the ECM filter method. Curves represent genes evaluated by patient age against gene expression level.

sostenible de ciudades inteligentes: movilidad urbana, datos abiertos, sensores moviles. SA070U 16. Project cofinanced with Junta Castilla y León, Consejería de Educación and FEDER funds.

Authors contributions

JR and JACG designed the proposed framework supervised by JFdP and JMC. JACG implemented the framework while JR provided biological background. JACG and JR wrote the paper whereas JFdP and JMC provided the comments and the discussion. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Appendix. Visualizations supporting the results

See Figs. A.1–A.5.

References

Ahdesmäki, A., Strimmer, K., 2010. Feature selection in omics prediction problems using CAT scores and false non-discovery rate control. Ann. Appl. Stat. 4, 503–519.

Ahdesmaki, M., Zuber, V., Gibb, S., Strimmer, K., 2015. sda: Shrinkage Discriminant Analysis and CAT Score Variable Selection. R package version 1.3.7, http://CRAN.R-project.org/package = sda.

Ambroise, C., McLachlan, G., 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc. Nat. Acad. Sci. U.S.A. (PNAS) 99, 6562–6566.

Badea, L., Herlea, V., Olimpia, S., Dumitrascu, T., Popescu, I., 2008a. Combined Analysis of Whole-Tissue and Microdissected PDAC. Bioinformatics group, National Institute for Research in Informatics Bucharest 011455, Romania.

Badea, L., Herlea, V., Olimpia, S., Dumitrascu, T., Popescu, I., 2008b. Combined gene expression analysis of whole-tissue and microdissected pancreatic ductal adenocarcinoma identifies genes specifically overexpressed in tumor epithelia. Hepato-Gastroenterology 88, 2015–2026.

Berrar, D.P., Dubitzky, W., Granzow, M., 2003. A Practical Approach to Microarray Data Analysis. Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow.

Bhaw-Luximon, A., Jhurry, D., 2015. New avenues for improving pancreatic ductal adenocarcinoma (PDAC) treatment: Selective stroma depletion combined with nano drug delivery. Cancer Lett. 369 (2), 266–273.

Bø, T., Jonassen, I., 2002. New feature subset selection procedures for classification of expression profiles. Genome Biology 4 (4), research0017.1–research0017.11.

Bourne, P., Wissig, H., 2003. Structural Bioinformatics. Wiley-Liss, Inc., Hoboken, New Jersey.

Castellanos-Garzón, J.A., 2012. Evolutionary Framework for DNA Microarray Cluster Analysis. (Ph.D. thesis), Department of Computer Science, University School of Computer Science, University of Valladolid.

Castellanos-Garzón, J.A., García, C., Novais, P., Díaz, F., 2013. A visual analytics framework for cluster analysis of DNA microarray data. In: Expert Systems with Applications, Vol. 40. Elsevier, pp. 758–774.

Castellanos-Garzón, J.A., Ramos, J., González-Briones, A., de Paz, J., 2016. A clustering-based method for gene selection to classify tissue samples in lung cancer.

In: Mohamad, M.S., et al. (Eds.), 10th International Conference on PACBB, Advances in Intelligent Systems and Computing, Vol. 477. Springer, pp. 99–107.

Castellanos-Garzón, J.A., Díaz, F., 2012. Clustergas: A hierarchical clustering method based on genetic algorithms. Technical Report CRAN R-Project. Department of Computer Science, University of Valladolid (Spain), http://cran.r-project.org/web/packages/clustergas. doi:http://cran.r-project.org/web/packages/clustergas R package version 1.0.

Castellanos-Garzón, J.A., Díaz, F., 2013. An evolutionary computational model applied to cluster analysis of DNA microarray data. In: Expert Systems with Applications, Vol. 40. Elsevier, pp. 2575–2591.

Chan, Z., Kasabov, N., 2004. Gene trajectory clustering with a hybrid genetic algorithm and expectation maximization method. In: IEEE International Joint Conference on Neural Networks, Vol. 3, pp. 1669–1674.

Chipman, H., Tibshirani, R., & with TSVQ code originally from Trevor Hastie 2006. hybridHclust: Hybrid hierarchical clustering. URL http://ace.acadiau.ca/math/chipmanh/hybridHclust R package version 1.0-1.

Crnogorac-Jurcevic, T., Chelala, C., Barry, S., Harada, T., Bhakta, V., Lattimore, S., Jurcevic, S., Bronner, M., Lemoine, N.R., Brentnall, T.A., 2013. Molecular analysis of precursor lesions in familial pancreatic cancer. Plos One 8 (1), e54830.

Deng, L., Pei, J., Ma, J., Lun, D., 2004. A rank sum test method for informative gene discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), pp. 410–419.

Díaz-Uriarte, R., Alvarez, S.D., 2006. Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7, 1–3.

Ding, C., Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. 29 (1), 185–205.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Statist. 32 (2), 407–499.

Eisen, M., Spellman, T., Brown, P., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Nat. Acad. Sci. U.S.A. 95, 14863–14868.

Flach, P., 2012. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press.

Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. Evol. Comput. 3, 1–16.

Geman, D., d'Avignon, C., Naiman, D., Winslow, R., 2003. Classifying gene expression profiles from pairwise mRNA comparisons. Stat. Appl. Genet. Mol. Biol. 3, 1–19.

Geoffrey, J., Do, K., Ambroise, C., 2004. Analyzing Microarray Gene Expression Data. John Wiley & Sons, Inc., Hoboken, New Jersey.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman, Inc..

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E., 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286 (5439), 531–537.

Guyon, I., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Han, J., Kamber, M., 2006. In: Gray, J. (Ed.), Data Mining: Concepts and Techniques. Elsevier Inc..

Haupt, R.L., Haupt, S.E., 2004. Practical Genetic Algorithms, second ed. Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Haury, A.-C., Gestraud, P., Vert, J.-P., 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE 6 (12), e28210.

Hernandez, J.C.H., Duval, B., Hao, J.-K., 2007. A genetic embedded approach for gene selection and classification of microarray data. In: EvoBIO 2007. In: Lecture Notes in Computer Science (LNCS), vol. 4447, Springer-Verlag, Berlin Heidelberg, pp. 90–101.

Hezel, A., Kimmelman, A., Stanger, B., Bardeesy, N., DePinho, R., 2006. Genetics and biology of pancreatic ductal adenocarcinoma. Genes & Dev. 20, 1218–1249.

Holland, J.H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press Edition.

Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A., 2004. Filter versus wrapper gene selection approaches in DNA microarray domains. In: Artificial Intelligence in Medicine, Vol. 31. Elsevier, pp. 91–103.

Ishwaran, H., Rao, J., 2005. Spike and slab variable selection: frequentist and bayesian strategies. Ann. Statist. 33 (2), 730–773.

Ishwaran, H., Rao, J., Kogalur, U.B., 2013. Spikeslab: Prediction and variable selection using spike and slab regression. R-package 1.1.5, http://web.ccs.miami.edu/hishwaran, http://www.kogalur.com.

Jaeger, J., Sengupta, R., Ruzzo, W., 2003. Improved gene selection for classification of microarrays. Pac. Symp. Biocomput. 8, 53–64.

Jager, J., Sengupta, R., Ruzzo, W., 2003. Improved gene selection for classification of microarrays. In: Pacific Symposium on Biocomputing (UW CSE Computational Biology Group), PMID: 12603017.

Jain, A.K., Dubes, R.C., 1998. In: Marttine, B. (Ed.), Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, New Jersey, p. 07632.

Jiang, D., Tang, C., Zhang, A., 2004. Cluster analysis for gene expression data: A survey. IEEE Trans. Knowl. Data Eng. 16 (11), 1370–1386.

Jolliffe, I.T., 2002. Principal Component Analysis. Springer-Verlag.

Kaufman, L., Rousseeuw, P.J., 2005. Finding Groups in Data. An Introduction to Clustering Analysis. John Wiley & Sons, Inc., Hoboken, New Jersey.

Keim, D.A., 2002. Information visualization and visual data mining. IEEE Trans. Vis. Comput. Graphics 8, 1–8.

Koorstra, J., Hustinx, S., Offerhaus, G., Maitra, A., 2008. Pancreatic carcinogenesis. Pancreatology 8 (2), 110–125.

Kumari, B., Swarnkar, T., 2011. Filter versus wrapper feature subset selection in large dimensionality microarray: A review. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) 2 (3), 1048–1053.

Kursa, M., Rudnicki, W., 2010. Feature selection with the Boruta package. J. Stat. Softw. 36 (11), 1–13.

Kursa, M., Rudnicki, W., 2016. Wrapper Algorithm for All Relevant Feature Selection. Package Boruta, Version 5.1.0, https://m2.icm.edu.pl/boruta/.

Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., deSchaetzen, V., Duque, R., Bersini, H., Nowé, A., 2012. A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (4), 1106–1118.

Liss, A., Thayer, S., 2012. Pancreatic cancer and tumor microenvironment. In: Grippo, P.J., Munshi, H.G. (Eds.), The Robert H. Lurie Comprehensive Cancer Center of Northwestern University Chicago. Transworld Research Network, Trivandrum (India), (Chapter 9).

Liu, X., Krishnan, A., Mondry, A., 2005. An entropy-based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6 (76), 1–14.

Long, A., Mangalam, H., Chan, B., Tolleri, L., Hatfield, G., Baldi, P., 2001. Improved statistical inference from dna microarray data using analysis of variance and a bayesian statistical framework. J. Biol. Chem. 276 (23), 19937–19944.

Macnaughton-Smith, P., Williams, W.T., Dale, M.B., Mockett, L.G., 1965. Dissimilarity analysis: a new technique of hierarchical subdivision. Nature 202, 1034–1035.

Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z., Metodiev, M., Lausen, B., 2014. A feature selection method for classification within functional genomics experiments based on the proportional overlapping score. BMC Bioinformatics 15 (274), 1–20.

Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z., Lausen, B., 2015. PropOverlap: Feature (gene) selection based on the Proportional Overlapping Scores. R package version 1.0, http://CRAN.R-project.org/package=propOverlap.

McDonald, J., 2014. Handbook of Biological Statistics, third ed.. Sparky House Publishing, Baltimore, Maryland.

Mohamed, A., Saberi, M., Deris, S., Omatu, S., Fdez-Riverola, F., Corchado, J., 2015. Gene knockout identification for metabolite production improvement using a hybrid of genetic ant colony optimization and flux balance analysis. In: Biotechnology and Bioprocess Engineering, Vol. 20. Springer, pp. 685–693.

Moorthy, K., Saberi, M., 2012. Random forest for gene selection and microarray data classification. In: Knowledge Technology, Third Knowledge Technology Week, KTW, Communications in Computer and Information Science, Vol. 295. Springer-Verlag, Berlin Heidelberg, pp. 174–183.

Natarajan, A., Ravi, T., 2014. A survey on gene feature selection using microarray data for cancer classification. Int. J. Comput. Sci. & Commun. (IJCSC) 5 (1), 126–129.

Nguyen, T., Khosravi, A., Creighton, D., Nahavandi, S., 2015. Hierarchical gene selection and genetic fuzzy system for cancer microarray data classification. PLos One 3 (10), 1–23.

Nicolai, S., Rossi, A., Di-Daniele, N., Melino, G., Annicchiarico-Petruzzelli, M., Raschella, G., 2015. DNA repair and aging: the impact of the p53 family. AGING 7 (12), 1050–1065.

Olson, D.L., Delen, D., 2008. Advanced Data Mining Techniques. Springer-Verlag, Berlin Heidelberg.

Pappa, G., Freitas, A., Kaestner, C., 2002. A multiobjective genetic algorithm for attribute selection. In: The Fourth International Conference on Recent Advances in Soft Computing. (RASC-2002), Springer, Berlin, pp. 116–121.

Peddada, S., Lobenhofer, E., Li, L., Afshari, C., Weinberg, C., Umbach, D., 2003. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19 (7), 834–841.

Penfold, C., Wild, D., 2011. How to infer gene networks from expression profiles, revisited. Interface Focus 1 (6), 857–870.

Quinlan, J., 1994. C4.5: Programs for machine learning. In: Machine Learning, Vol. 16. Springer, pp. 235–240.

R Core Team 2015. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna, Austria. URL https://www.R-project.org/.

Ruiz, R., Riquelme, J., Aguilar-Ruiz, J., 2006. Incremental wrapper-based gene selection from microarray data for cancer classification. In: Pattern Recognition, Vol. 39. Elsevier, pp. 2383–2392.

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

Schroeder, M., Gilbert, D., Helden, J.V., Noy, P., 2001. Approaches to vusualisation in bioinformatics: from dendrograms to space explorer. In: Information Sciences, Vol. 139. Elsevier, pp. 19–57.

Shraddha, S., Anuradha, N., Swapnil, S., 2014. Feature selection techniques and microarray data: A survey. Int. J. Emerg. Technol. Adv. Eng. 4 (1), 179–183.

Simeka, K., Fujarewicza, K., Swierniaka, A., Kimmela, M., Jarzab, B., Wienchc, M., Rzeszowskac, J., 2004. Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data. Eng. Appl. Artif. Intell. 17, 417–427.

Speed, T., 2003. In: Speed, T. (Ed.), Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC Press LLC.

Tan, P., Steinbach, M., Kumar, V., 2006. Introduction to Data Mining. Addison-Wesley.

Thomas, J., Olson, J., Tapscott, S., Zhao, L., 2001. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Res. 11 (7), 1227–1236.

TunedIT, S., 2008. Machine learning & data maning algorithms. automated tests, repeatable experiments, meaningful resutls. http://tunedit.org/challenge/rsctc-2010-b. Academic Technology Incubator, University of Warsaw.

Tusher, V., Tibshirani, R., Chu, G., 2001. Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. 98 (9), 5116–5121.

Tyagi, V., Mishra, A., 2013. A survey on different feature selection methods for microarray data analysis. Int. J. Comput. Appl. 67 (16), 36–40.

Wang, Y., Tetko, I., Hall, M.A., Frank, E., Facius, A., Mayer, K.F., Mewes, H.W., 2005. Gene selection from microarray data for cancer classification - a machine learning approach. In: Computational Biology and Chemistry, Vol. 29. Elsevier, pp. 37–46.

Weiss, P., 2005. Applications of generating functions in nonparametric tests. Math. J. 9 (4), 803–823.

Wolters, M., 2015a. A Genetic Algorithm for Fixed-Size Subset Selection. R-Package kofnGA, Version 1.2.

Wolters, M., 2015b. A genetic algorithm for selection of fixed-size subsets with application to design problems. J. Stat. Softw. 68 (1), 1–18.

Xing, E., Jordan, M., Karp, R., 2001. Feature selection for high-dimensional genomic microarray data. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML'01, pp. 601–608.

Yang, K., Cai, Z., Li, J., Lin, G., 2006. A stable gene selection in microarray data analysis. BMC Bioinformatics 7 (228), 1–16.

Yee, K., Bumgarner, R., Raftery, A., 2005. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. Bioinformatics 21 (10), 2394–2402.

Yeung, K., Bumgarner, R., 2003. Multiclass classification of microarray data with repeated measurements: Application to cancer. Genome Biol. 4 (12), R83.

Zhou, X., Tuck, D., 2007. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. Bioinformatics 23 (9), 1106–1114.