

UNIVERSITÀ DEGLI STUDI DI MILANO



SCUOLA DI DOTTORATO

Scienze biomediche cliniche e sperimentali

DIPARTIMENTO

Dipartimento di Scienze Cliniche e di Comunità

CURRICULUM / CORSO DI DOTTORATO

Statistica biomedica - ciclo XXVIII

TESI DI DOTTORATO DI RICERCA

**A COMPREHENSIVE PIPELINE FOR
CLASS COMPARISON AND CLASS PREDICTION IN CANCER RESEARCH**

SETTORE SCIENTIFICO-DISCIPLINARE

MED/01

CANDIDATO: DR. ELENA LANDONI

TUTOR: DR. ROSALBA MICELI

CO-TUTOR: DR. FEDERICO AMBROGI

COORDINATORE: PROF. ADRIANO DECARLI

A.A. 2014/2015

Contents

Abstract	v
Acknowledgements	vi
1. Introduction	1
1.1 The EDERA project	1
1.2 Past and present research activities	1
2. Class comparison analysis	2
2.1 Location tests	4
2.1.1 Student's and Welch's t tests	4
2.1.2 Wilcoxon-Mann-Whitney test	5
2.2 Tests for the location-scale problem	6
2.2.1 Podgor-Gastwirth PG2 test	6
2.2.2 Cucconi test	7
2.3 Tests for the general two-sample problem	8
2.3.1 Kolmogorov-Smirnov test	8
2.3.2 Cramer-von Mises test	9
2.3.3 Anderson-Darling test	9
2.3.4 Zhang tests	10
2.4 Simulation study on parametric and nonparametric two-sample tests for feature screening ...	12
2.4.1 Simulation settings	12
2.4.2 Simulation results	14
2.4.3 Real data example	16
2.4.4 Final remarks	19
2.5 Combined use of t and Anderson-Darling tests	20
3. Class prediction analysis	22
3.1 First step: feature selection	22
3.1.1 Prediction Analysis for Microarrays	22
3.1.2 Random Forest	24
3.1.3 Support Vector Machines	30
3.2 Second step: classifier development	39
3.2.1 Linear Support Vector Machines	40
3.2.2 Cross Validation procedure	42
3.3 Presentation of results	44
3.3.1 The 'egg-shaped' plot	44
3.3.2 The 'ROC space' plot	45

3.4 The Cineca system	46
4. A comprehensive pipeline for class comparison and class prediction	47
5. Real data applications.....	48
5.1 Plasmatic microRNA data	48
5.2 Secondary ElectroSpray Ionization-Mass Spectrometry data	61
6. Conclusions	72
7. References	74
7.1 References for plasmatic microRNA data.....	79
7.2 References for Secondary ElectroSpray Ionization-Mass Spectrometry data	82
Appendix 1: R codes for Cramer-von-Mises - Podgor-Gastwirth PG2 - Cucconi - Zhang tests	85
Appendix 2: Density distributions and Empirical Cumulative Distribution Functions of the simulation patterns I-VI	89
Appendix 3: Simulation results in terms of size and power ($\alpha = 0.05$)	109
Appendix 4: Real data example: exemplificative kernel density distributions of selected 'gray-zone' genes in ER positive (in red) and ER negative (in green) subjects.....	117

Abstract

Personalized medicine is an emerging field that promises to bring radical changes in healthcare and may be defined as “a medical model using molecular profiling technologies for tailoring the right therapeutic strategy for the right person at the right time, and determine the predisposition to disease at the population level and to deliver timely and stratified prevention”. The sequencing of the human genome together with the development and implementation of new high throughput technologies has provided access to large ‘omics’ (e.g. genomics, proteomics) data, bringing a better understanding of cancer biology and enabling new approaches to diagnosis, drug development, and individualized therapy. ‘Omics’ data have the potential as cancer biomarkers but no consolidated guidelines have been established for discovery analyses. In the context of the EDERA project, funded by the Italian Association for Cancer Research, a structured pipeline was developed with innovative applications of existing bioinformatics methods including: 1) the combination of the results of two statistical tests (t and Anderson-Darling) to detect features with significant fold change or general distributional differences in class comparison; 2) the application of a bootstrap selection procedure together with machine learning techniques to guarantee result generalizability and study the interconnections among the selected features in class prediction. Such a pipeline was successfully applied to plasmatic microRNA, identifying five hemolysis related microRNAs and to Secondary ElectroSpray Ionization-Mass Spectrometry data, in which case eight mass spectrometry signals were found able to discriminate exhaled breath from breast cancer patients from that of healthy individuals.

Acknowledgements

Dedico la tesi alle persone che mi hanno trasmesso forza e positività in questi tre anni di Dottorato.

Vorrei iniziare a ringraziare mia mamma, Nonna Maria e zii Renato e Vania per il costante sostegno.

Francesco, persona speciale che con affetto e pazienza mi è rimasta accanto, anche nei momenti più difficili.

Marta, amica dal cuore grande e compagna di avventure (sei sempre in giro e ti vedo raramente, ma è sempre come se ci vedessimo il giorno prima).

Elisa, amica 'di vecchia data', con cui ho condiviso molti pensieri e confidenze.

Alessandra, novella sposina e compagna di pazzie.

Otty, sorellina dalle mille risorse e fan dell'alpaca.

Francesca, personalità geniale in giro per il mondo, che spero presto di riabbracciare.

Antonio ('cugino'), favoloso sublocatore milanese.

Marica, per la sensibilità e capacità di ascolto.

Gabriele, per il costante buonumore e per le ormai classiche 'gabrizzate'.

Alba, tutor presente che mi ha costantemente spronato a fare meglio e supportato (o meglio sopportato) in questi anni.

Federico (Iodigiano), prezioso co-tutor che mi ha introdotto nel magico mondo Cineca.

Luigi, per le chiacchierate la mattina presto e per il continuo supporto.

Adriano, per il senso dell'umorismo e i sorrisi immancabili.

Chiara (grande), amica preziosa e fonte inesauribile di buoni consigli.

Marco, per le canzoni cantate insieme e i continui solleciti al silenzio (!).

Patrizia, per aver condiviso con me la passione per la letteratura inglese e per le pause pomeridiane.

Chiara (piccola), per il senso pratico e i piccoli rimproveri al momento giusto.

Loredana, per i caffè e la compagnia mattutini, specialmente nel periodo sotto tesi.

Annalisa, Sara, Stefano, Mara, Maddalena, Paolo, Ilaria, Monica, Giuseppe, Gabriele, Federico (siciliano), Delphine, Giò, Tiziana, Valentina, Teresa, Alessandra, Matteo, Valeria, Francesca, Rosaria, Salvatore, per le risate e i momenti passati insieme.

A tutti un forte abbraccio e un GRAZIE di cuore.

“Ma un giorno, così, succede che inizi a camminare sentendo i tuoi passi, e a sentire la vita intorno, e anche gli occhi delle persone sembrano più vicini. Senti che ci sei, libera e fiera nel tuo incedere, quasi fosse interminabile.”

1. Introduction

1.1 The EDERA project

The present research is integrated within a multidisciplinary project funded by the Italian Association for Cancer Research (AIRC), *i.e.* the EDERA (Early DEtection and Risk Assesment) project [1]. The project title is ‘Tumor microenvironment-related changes as new tools for early detection and assessment of high-risk diseases’ and it is aimed at investigating the following topics: 1. relevance of bone marrow-related cells and molecules for early diagnosis and risk assessment; 2. diagnostic value of signals from tumor microenvironment and extracellular matrix; 3. tumor-microenvironment interactions and genetic risk; 4. clinical impact of tumor-microenvironment related changes. Therefore this project will provide a proof-of-concept that early stroma modifications in the tumor and in its microenvironment have biological and clinical relevance for the early detection of cancer lesions and for the early identification of aggressive tumors. From a statistical point of view, the aim is to construct and validate diagnostic, prognostic and predictive blood and tissue-based biomarkers; this can be useful for diagnosing cancer in its earliest stages, diagnosing cancer relapses, or indicating sensitivity to molecular drugs.

1.2 Past and present research activities

Because of the complexity of EDERA project purposes, several types of ‘omics’ (*e.g.* genomics and proteomics) data were analyzed. Specifically, during the three years of the Ph.D., we examined plasmatic microRNAs (miRNAs), Liquid Chromatography-Mass Spectrometry (LC-MS) data, immunohistochemistry (IHC), Fluorescent Activated Cell Sorting (FACS) data and Enzyme-Linked ImmunoSorbent Assay (ELISA) data. This research area is characterized by data with a huge number of variables (herein indicated as ‘features’) assessed in small samples and, from a more general point of view, can be seen as high dimensional data. This issue encouraged us to use traditional statistical methods together with ones specifically addressed to high dimensional data. The Unit for Statistical Coordination of the Studies I work in provides methodological and practical support to the different Work Packages (WPs) involved in the EDERA project. During the first two years of the Ph.D., besides performing applied statistical analyses together with methodological insights, I mainly focused on class comparison analysis and set up a simulation study on parametric and nonparametric two-sample tests for feature screening. We considered the following tests: Student’s t , Welch’s t , Wilcoxon-Mann-Whitney (WMW), Kolmogorov-Smirnov (KS), Anderson-Darling (AD), Cramér-von Mises (CvM), Podgor-Gastwirth PG2, Cucconi and Zhang tests (Z_K , Z_C and Z_A) (chapter 2). During the third year of Ph.D., I focused on class prediction

analysis, both methodologically (chapter 3) and with real data applications (chapter 5). The final result of my Ph.D. project consists in a comprehensive pipeline for class comparison and class prediction analyses (chapter 4), flexible enough to adapt to different types of ‘omics’ data. In detail, my research activity is summarized in the following Table.

Area	Aim/Issue	Statistical method/Programming tool
Data pre-processing	Normalization with circulating miRNA data	Ratio-based normalization method [2]
	Batch effect correction	ComBat method [3]
	Balancing groups in the design phase	Caliper Propensity Score Matching [4]
	Concordance analysis	Concordance correlation coefficient [5]
	Optimal cutoff search	Index Score [6]
Class comparison	Under-detected MS data	Tobit model [7]
	Search for differentially expressed (DE) features	Tests for the general two-sample problem [8]
	Feature ranking in survival analysis	Generalized Boosted Model (GBM) [9]
Class prediction	Feature selection and classifier development	Two-step strategy: bootstrap feature ranking with three algorithms - Prediction Analysis for Microarrays, Random Forest, Elastic Smoothly Clipped Absolute Deviation Support Vector Machines (SVM) + classifier development with Linear (SVM) model [10]
Miscellaneous analyses	Association studies with non normal data	Quantile regression model [11]
	Computational time reduction	Parallel programming
	Reproducible research: dependence of results on pseudorandom seeds	Random Number Generator (RNG) [12]

2. Class comparison analysis

Class comparison analysis is mainly directed to find the features differentially expressed (DE) between two conditions. A significance threshold above which a feature is declared differentially expressed has to be established. To this aim, statistical tests are needed, requiring the definition of appropriate test statistics and the control of the level of the tests.

Parametric and nonparametric two-sample tests are applied in a large number of high-dimensional continuous data for explorative studies in order to detect DE features (genes, miRNAs, metabolites) between different biomedical conditions, such as diseased (cases) and healthy subjects (controls). In particular, the tests are exploited as univariable feature ranking methods in class comparison ([13]), as well as a preliminary step - feature screening- in class prediction ([14]). Such a screening is intended to identify promising features to be possibly included in a multivariable model, referred to as the predictor or classifier, which aims at accurately predicting the class membership of a new sample based on a combination of expression levels of the selected features. The most commonly used tests are the t tests [15] and the nonparametric Wilcoxon-Mann-Whitney test [16], which refer to differences in terms of

location and therefore are classified as location tests. However, feature distributions in the comparison classes may differ according to other aspects such as scale or, more generally, shape. One could test for location or scale changes (location-scale problem) or look for any changes in location, scale or shape (general two-sample problem) ([17]). Even small signals of general differences between the two classes could reveal discriminative features that should not be filtered out in the first phases of bioinformatics analyses, but further investigated in the following step of class prediction. Moreover, the parametric test assumption of normality is often not fulfilled when dealing with some types of genomic data, mainly due to the small sample size ([18]), producing skewed, heavy-tailed or multimodal distributions of expression values. In presence of such distributions, nonparametric alternatives to location tests *e.g.* the Kolmogorov-Smirnov filter proposed by [19], could be more sensitive in feature screening, thus leading to a small number of false negative discoveries. In the field of high dimensional data, feature screening should not be tailored on specific distributional characteristics but rather be a flexible procedure, *i.e.* able to detect general differences between feature distributions under different patterns. Thus, a desirable test should prove to be robust in terms of Type I error control and powerful in a wide family of distributional patterns, even if not being the best one in every single situation. Previous studies compared via simulation limited ([17]) or specific (*i.e.* directed to show a specific type of difference ([20]; [21])) sets of two-sample tests. The above scenario prompted us to conduct an extensive simulation study with small sample sizes and non normal distributions, involving a wide set of parametric and nonparametric tests for two class comparison, which were compared according to their size (*i.e.* type I error rate) and power, with the aim of possibly identifying a test to be used in the screening phase of high dimensional explorative studies. We investigated a series of nonparametric tests considering different alternatives versus the null hypothesis of equality between the Cumulative Distribution Functions (CDFs). Being aware that when the parametric tests assumptions are violated their power is deflated, our aim was to assess possible nonparametric alternatives and comparatively draw indications of possible improvements over the parametric tests. In particular, we implemented the following nonparametric tests:

- the Wilcoxon-Mann-Whitney (WMW) test, detecting shifts in location between the CDFs;
- two tests for the location-scale problem, *i.e.* the PG2 Podgor-Gastwirth (PG2) [22]) and the Cucconi test ([20]; [23]); the PG2 test has been recognized as the most powerful among the PG efficiency robust tests, while the Cucconi test represents the simplest and best performing alternative to PG2 ([20]);

- three chi-squared statistic-based tests, *i.e.* the Kolmogorov-Smirnov (KS) ([24]), the Cramer-von Mises (CvM) ([25]) and the Anderson-Darling (AD) ([26]) test; the KS test refers to the CDF maximum difference, the CvM test considers differences over the entire CDF range, while the AD test takes into account global CDF differences, granting more importance to the observations in the tails; the latter characteristic makes the AD test valuable when one is interested in finding also signals that are only present in a subset of patients;
- the Zhang Z_K , Z_C and Z_A tests, which are 'likelihood-ratio' based analogs of the 'traditional' KS, CvM and AD tests, respectively ([17]).

As regards the simulation study, we chose to mimic the irregular pattern of the feature distributions of the two samples by using mixtures of two normal distributions (NM), which should reproduce the coexisting presence of heterogeneous subpopulations underlying data, by varying the mixture parameters. In paragraphs 2.1, 2.2 and 2.3, we briefly describe the considered tests; the simulation settings and results (*i.e.* the comparison of the tests in terms of size and power under different distributional patterns) are reported in paragraph 2.4. In paragraph 2.5 it is illustrated the combined use of t and AD tests, with an application to plasmatic miRNA data (described in paragraph 5.1).

We can classify the investigated tests into three categories: I. location tests (Student's t, Welch's t, WMW), II. tests for the location-scale problem (PG2, Cucconi) and III. tests for the general two-sample problem (KS, CvM, AD, Z_K , Z_C , Z_A).

In the following, let x_1, \dots, x_m and y_1, \dots, y_n be the observations drawn from two independent random variables X and Y with continuous CDFs $F(x)$ and $G(y)$, respectively.

2.1 Location tests

The location tests assess whether the center of the data is the same for the two distributions.

2.1.1 Student's and Welch's t tests

Let μ_1 and μ_2 be the means and σ_1^2 and σ_2^2 be the variances of the random variables X and Y. The null and alternative hypotheses of the considered parametric tests are:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

The t test is defined as

$$t = \frac{\bar{X} - \bar{Y}}{s \sqrt{\frac{1}{m} + \frac{1}{n}}} \sim t_{m+n-2}$$

and assumes that $\sigma_1^2 = \sigma_2^2$, thus estimating the pooled sample variance as

$$s^2 = \frac{1}{m+n-2} \left(\sum_{i=1}^m (X_i - \bar{X})^2 + \sum_{j=1}^n (Y_j - \bar{Y})^2 \right)$$

The Welch's t variant (Satterthwaite -Welch adjustment) assumes $\sigma_1^2 \neq \sigma_2^2$, and is defined as

$$Welch = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}} \sim t_\nu \quad \nu \approx \frac{\left(\frac{s_1^2}{m} + \frac{s_2^2}{n}\right)^2}{\frac{\left(\frac{s_1^2}{m}\right)^2}{m-1} + \frac{\left(\frac{s_2^2}{n}\right)^2}{n-1}}$$

Both the tests were performed using the *t.test* function included in the *stats* package.

2.1.2 Wilcoxon-Mann-Whitney test

The WMW test considers shifts in location between the two CDFs:

$$H_0 : F(x) = G(y) \quad \forall x, y$$

$$H_1 : G(y) = F(x - \Delta) \quad (\Delta > 0 \text{ or } \Delta < 0)$$

Let R_1 and R_2 be the sums of the ranks for the observations in the two groups, $U = \min(U_1, U_2)$, where $U_1 = R_1 - [m(m+1)/2]$ and $U_2 = R_2 - [n(n+1)/2]$.

For $m \geq 8$ and $n \geq 8$, a normal approximation is used to calculate the standardized WMW test statistic:

$$WMW = \frac{U - \mu_u}{\sigma_u} \quad \mu_u = \frac{mn}{2} \quad \sigma_u = \sqrt{\frac{mn(N+1)}{12}}$$

The WMW test considers the differences between the two distributions in terms of shifts in location, *i.e.* the null hypothesis is rejected if there is a prevalence of high ranks (or low ranks) in one group. The test was performed using the *wilcox.test* function included in the *stats* package.

2.2 Tests for the location-scale problem

The tests for the location-scale problem assess whether both the samples come from the same distribution:

$$H_0 : F(x) = G(y) \quad \forall x, y$$

against the location-scale alternative hypothesis:

$$H_1 : G(y) = F\left(\frac{x - \mu}{\sigma}\right)$$

with $\mu \neq 0$ or $\sigma \neq 1$.

2.2.1 Podgor-Gastwirth PG2 test

Let I_i , $i = 1, \dots, N$ be a group indicator so that $I_i = 1$ when the i^{th} element of the combined sample belongs to the first sample, $I_i = 0$ otherwise; let S_i and S_i^2 be the ranks and the squared ranks of the observations in the combined sample ($i=1, \dots, N$). The PG2 test statistic is calculated as the ordinary least squares (OLS) estimator of S_i and S_i^2 on the group indicators I_i and is distributed as a Fisher-Snedecor F with 2 and $N-3$ degrees of freedom:

$$PG2 = \frac{(\mathbf{b}^T \mathbf{S}^T \mathbf{I} - m^2/N)/2}{(m - \mathbf{b}^T \mathbf{S}^T \mathbf{I})/(N - 3)} \sim F_{2, N-3}$$

In the above formula, T denotes the transpose operator, \mathbf{b} is the 3 x 1 vector of the OLS estimate of the intercept term and the regression coefficients, \mathbf{S} the $N \times 3$ matrix of the ranks and the squared ranks of the observations and \mathbf{I} the $N \times 1$ vector of the group indicators I_1, \dots, I_N :

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 1 & S_1 & S_1^2 \\ 1 & S_2 & S_2^2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & S_N & S_N^2 \end{bmatrix} \quad \mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ I_N \end{bmatrix}$$

Large values of the statistic imply the rejection of H_0 . Podgor and Gastwirth showed that asymptotically the PG2 test can be recast as a quadratic combination of the Wilcoxon rank test for location and the Mood squared rank test for scale (Mood's scores give more weight to the extreme ranks).

2.2.2 Cucconi test

The Cucconi test addresses the location-scale problem by using the squares of ranks (S_i) and contrary-ranks ($N+1 - S_i$) of the observations of the sample X_i ($i=1, \dots, m$) computed in the pooled sample. The test statistic is defined as:

$$C = \frac{U^2 + V^2 - 2\rho UV}{2(1 - \rho^2)}$$

where:

$$U = \frac{6 \sum_{i=1}^m S_i^2 - m(N+1)(2N+1)}{\sqrt{mn(N+1)(2N+1)(8N+11)/5}}$$

$$V = \frac{6 \sum_{i=1}^m (N+1 - S_i)^2 - m(N+1)(2N+1)}{\sqrt{mn(N+1)(2N+1)(8N+11)/5}}$$

$$\rho = \frac{2(N^2 - 4)}{(2N+1)(8N+11)} - 1$$

Note that U is based on the squares of the ranks S_i , while V is based on the squares of the contrary-ranks ($N+1 - S_i$) of the first sample. Let U' and V' be U and V computed referring to the second sample Y_j ($j=1, \dots, n$); the aforementioned expressions of U and V become, respectively:

$$U' = \frac{6 \sum_{j=1}^n S_j^2 - n(N+1)(2N+1)}{\sqrt{mn(N+1)(2N+1)(8N+11)/5}}$$

$$V' = \frac{6 \sum_{j=1}^n (N+1 - S_j)^2 - n(N+1)(2N+1)}{\sqrt{mn(N+1)(2N+1)(8N+11)/5}}$$

Since

$$\sum_{i=1}^m S_i^2 + \sum_{j=1}^n S_j^2 = \sum_{i=1}^m (N+1 - S_i)^2 + \sum_{j=1}^n (N+1 - S_j)^2 = \frac{N(N+1)(2N+1)}{6}$$

then $U' = -U$ and $V' = -V$. Thus, the two test statistics are equal:

$$C' = \frac{U'^2 + V'^2 - 2\rho U'V'}{2(1 - \rho^2)} = \frac{U^2 + V^2 - 2\rho UV}{2(1 - \rho^2)} = C$$

It makes no difference whether U and V are computed based on the data of the first or the second sample, since this choice does not modify the test statistic. Large values of the statistic imply the rejection of H_0 . For the asymptotic Cucconi test we used the reported critical threshold of $-\ln(\alpha)$.

2.3 Tests for the general two-sample problem

Like the WMW test and the tests for the location-scale problem, the tests for the general two-sample problem assess whether both samples come from the same distribution:

$$H_0 : F(x) = G(y) \quad \forall x, y$$

However, the alternative hypothesis is:

$$H_1 : F(x) \neq G(y)$$

and thus it evaluates general differences between the two CDFs. The KS test concentrates on local CDF shifts while the CvM and AD tests consider the differences all along the CDF distribution.

2.3.1 Kolmogorov-Smirnov test

The KS test statistic is defined as the largest absolute value of the difference between the Empirical Cumulative Distribution Functions (ECDFs) of the two samples:

$$KS = \sup_x |F_m(x) - G_n(y)|$$

For large sample sizes, *i.e.* $m = n$ with $n > 40$ (balanced sample sizes) and $m > 16$ and $n > 20$ (unbalanced samples), the large sample approximation is used ([27]) and the null hypothesis is rejected at level α when:

$$KS > \frac{c(\alpha)}{\sqrt{n}}$$

for balanced sample sizes, or

$$KS > c(\alpha) \sqrt{\frac{m+n}{mn}}$$

for unbalanced sample sizes, where $c(\alpha)$ are tabulated values ([27] - Tables 16 and 17). The test was performed using the *ks.test* function included in the *stats* package.

2.3.2 Cramer-von-Mises test

The CvM test considers the difference between the two distributions over the entire CDF range. The L_2 -norm based version of the CvM test, which was introduced by Anderson et al. ([25]) and involves the quadratic distance between the two ECDFs, was considered. Let $H_N(x,y) = mF_m(x) + nG_n(y)$, being H_N the ECDF associated with the combined sample. Then the CvM test statistic is defined as:

$$CvM = \frac{mn}{N} \int_{-\infty}^{+\infty} |F_m(x) - G_n(y)|^2 dH_N(x, y)$$

which is equivalent to:

$$CvM = \frac{mn}{N^2} \left[\sum_{i=1}^m (F_m(x_i) - G_n(x_i))^2 + \sum_{j=1}^n (F_m(y_j) - G_n(y_j))^2 \right]$$

The null hypothesis is rejected for large values of CvM; asymptotic critical values are reported by Anderson and Darling [25]. However, the distance between the two ECDFs tends to 0 when $x \rightarrow -\infty$ or $x \rightarrow +\infty$, thus the value of the CvM test statistic is rather insensitive to the differences in the distribution tails. We performed the test by implementing a user-defined function including the empirical correction formula reported by Burr ([28]), using the limiting distribution to approximate the exact distribution of the CvM test statistic.

2.3.3 Anderson-Darling test

The AD test statistic is a modification of L_2 -CvM test statistic that, in order to give more weight to the observations in the distribution tails, includes a weighting function equal to the reciprocal of the variance of the ECDF (the latter is maximal around the median and minimal in the tails):

$$A_{mn}^2 = \frac{mn}{N} \int_{-\infty}^{+\infty} \frac{(F_m(x) - G_n(y))^2}{H_N(x, y)(1 - H_N(x, y))} dH_N(x, y)$$

A simplification was introduced for computational purposes:

$$A_{mn}^2 = \frac{1}{mn} \sum_{i=1}^{N-1} \frac{(M_i N - mi)^2}{i(N - i)}$$

where M_i is defined as the number of observations in the first sample less than or equal to the i^{th} smallest in the pooled sample. The standardized statistic is obtained by using its exact mean (equal to 1 in case of two samples), and exact variance σ_N , which was derived by Scholz ([26]):

$$AD = \frac{A_{mn}^2 - 1}{\sigma_N}$$

The upper tail critical values for the aforementioned test statistic are reported by Scholz ([26]) and the null hypothesis is rejected for large values. The standardization removes some of the dependence of the test on the sample size, as it was confirmed through a Monte Carlo study ([26]). For not tabulated critical values, an interpolation formula may be used to obtain the percentiles of interest. The test was performed using the *adk.test* function included in the *adk* package.

2.3.4 Zhang tests

All the considered Zhang tests (Z_K , Z_C , Z_A) derive from two types of test statistics ([29]) defined as:

$$Z = \int_{-\infty}^{+\infty} Z_t dw(t)$$

and

$$Z_{max} = \sup [Z_t w(t)] \quad t \in (-\infty, +\infty)$$

where Z_t is the likelihood ratio test statistic and $w(t)$ is a weighting function characterizing the different tests. The Zhang tests are the analogs of the traditional tests KS, CvM and AD, which are obtained using the Pearson χ^2 test statistic as Z_t .

Zhang Z_K test.

Z_K is the analog of the KS test and it is obtained from Z_{max} with $w(t) = 1$. The computational formula for the Z_K test statistic is:

$$Z_K = \max_{1 \leq k \leq N} \left[m \left(F_m \ln \frac{F_m}{H_N} + (1 - F_m) \ln \frac{1 - F_m}{1 - H_N} \right) + n \left(G_n \ln \frac{G_n}{H_N} + (1 - G_n) \ln \frac{1 - G_n}{1 - H_N} \right) \right]$$

where

$$F_m = F_m(x(k)); \quad G_n = G_n(y(k)); \quad H_N = H_N(x(k), y(k))$$

and H_N denotes the ECDF of the pooled sample ($k=1, \dots, N$). Large values of the statistic guide to the rejection of H_0 .

Zhang Z_C test.

Z_C is the analog of the CvM test and it is obtained from Z with $dw(t)$ defined as:

$$F(t)^{-1}[1 - F(t)]^{-1}dF(t)$$

where $F(t)$ is the common underlying distribution under H_0 . Let R_1 denote the rank in the pooled sample of the i^{th} -ordered statistic $X_{(i)}$ in the first sample X_i ($i=1, \dots, m$) and R_2 denote the rank in the pooled sample of the j^{th} -ordered statistic $Y_{(j)}$ in the second sample Y_j ($j=1, \dots, n$). The computational formula for the Z_C test statistic is:

$$Z_C = \frac{1}{N} \left[\sum_{i=1}^m \ln\left(\frac{m}{i-0.5} - 1\right) \ln\left(\frac{N}{R_1-0.5} - 1\right) + \sum_{j=1}^n \ln\left(\frac{n}{j-0.5} - 1\right) \ln\left(\frac{N}{R_2-0.5} - 1\right) \right]$$

Small values of the statistic guide to the rejection of H_0 .

Zhang Z_A test.

Z_A is the analog of the AD test and it is obtained from Z with $dw(t)$ defined as:

$$F_m(t)^{-1}[1 - F_m(t)]^{-1}dF_m(t)$$

The computational formula for the Z_A test statistic is:

$$Z_A = - \sum_{k=1}^N \left[m \frac{F_m \ln F_m + (1 - F_m) \ln(1 - F_m)}{(k - 0.5)(N - k + 0.5)} + n \frac{G_n \ln G_n + (1 - G_n) \ln(1 - G_n)}{(k - 0.5)(N - k + 0.5)} \right]$$

Small values of the statistic guide to the rejection of H_0 .

R codes of user-defined functions for the implementation of PG2, Cucconi, CvM and Zhang tests are reported in the Appendix 1.

2.4 Simulation study on parametric and nonparametric two-sample tests for feature screening

2.4.1 Simulation settings

A simulation study was conducted in order to compare the performance - in terms of size and power - of the tests previously described under different distributional patterns. Following the considerations made by Burton et al. ([30]), we decided to simulate data with some resemblance to real continuous high dimensional data, replicating their irregular distributional patterns by using mixtures of two normal distributions (NM) (Figure 1).

Let μ_{iA} and μ_{iB} be the means of the two components A and B of the mixture in the sample i ($i=1,2$), with $\mu_{iB} - \mu_{iA} = sh_i$ (shift), σ_{iA}^2 and σ_{iB}^2 be the mixture variances and λ_i the mixture weight, which is the probability associated with the first component of the mixture. Finally, let $\delta = \mu_2 - \mu_1$ be the difference between the two overall mixture means in the two samples, with $\mu_i = \lambda_i \mu_{iA} + (1 - \lambda_i) \mu_{iB}$. Three main cases were simulated: A. two normal distributions; B. one normal and one mixture distributions; C. two mixture distributions; a particular sub-case with equal shifts $sh_1 = sh_2 = sh$ was also considered. The parameters δ , sh_1 , sh_2 were properly tuned over fixed ranges in order to simulate the different conditions under H_0 and H_1 . We considered four mixture weights $\lambda \in \{0.80; 0.95; 0.20; 0.05\}$ and three small sample size settings, one balanced ($m=20$ vs $n=20$) and two unbalanced ($m=20$ vs $n=40$ and $m=40$ vs $n=20$) (Table 1). We fixed $\sigma_{1A}^2 = \sigma_{1B}^2 = \sigma_{2A}^2 = \sigma_{2B}^2 = 1$; this is not a limitation since the mixture overall means μ_1 and μ_2 will not be constant but will vary according to sh_1 and sh_2 , as well as the mixture overall variances σ_1^2 and σ_2^2 which will change according to shifts and mixture weights.

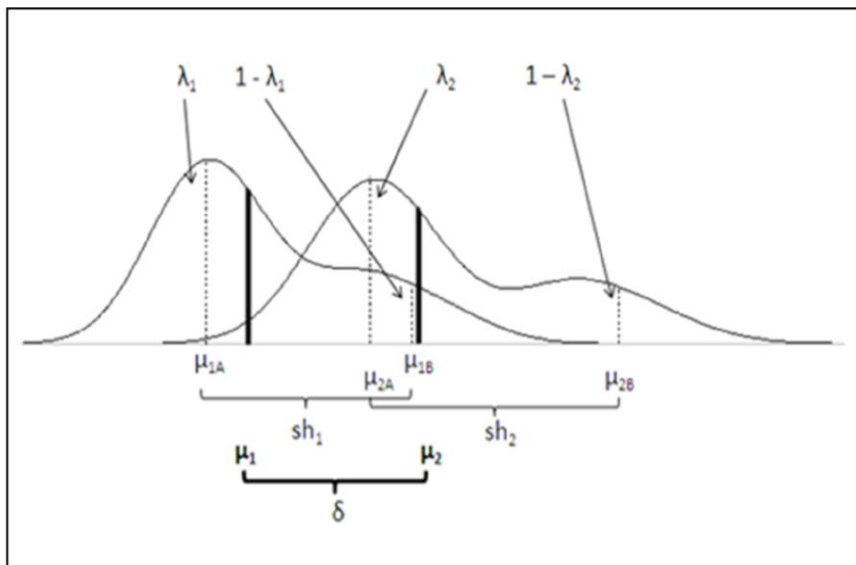


Figure 1. Example figure of two normal mixture distributions setting adopted into the simulation. μ_{iA} and μ_{iB} are the means of the two components A and B of the mixtures in the two samples i ($i=1,2$) with shifts $sh_i = \mu_{iB} - \mu_{iA}$ ($i=1,2$), $\delta = \mu_2 - \mu_1$ is the difference between the two mixture overall means and λ_i are the two mixture weights of the A components, being their complement the mixture weights of the respective B components.

Parameter	Values
λ	{0.8; 0.95; 0.2; 0.05}
δ	{0; 1}
sh	{0; 3; 6}
(m,n)	{(20,20); (20,40); (40,20)}

Table 1. Ranges of values for the parameters λ , δ , sh and (m,n).

We distinguished six different patterns, summarized in Table 2 and graphically represented in the Appendix 2 for each combination of sample size settings and mixture weights. We chose to consider a nominal significance level $\alpha = 0.05$ and to perform $B = 10000$ simulations so as to obtain precise estimates derived via the simulation. As an indicator of the simulation error we chose the standard error $SE(p)$, with p indicating the nominal coverage probability ([30]). Finally, we calculated the relative frequencies of H_0 rejection of the tests; under H_0 such frequencies are expected to approximate the fixed nominal significance level α , while under H_1 they correspond to the empirical power of the tests. It was possible to simulate the null hypothesis patterns only when comparing two normal distributions with equal means (pattern I) or two perfectly overlapping mixture distributions, *i.e.* those with equal shifts (pattern IV). The robustness of the tests under H_0 was evaluated according to the indications given by both Conover ([31]) and Marozzi ([32]): a test is considered robust if its Maximum Estimated Significance Level (MESL), *i.e.* its maximum relative frequency of H_0 rejection under H_0 (Table 2 - H_0 patterns) does not exceed a given threshold, typically 2α or 1.5α to be more restrictive.

H_0 patterns ($\delta = 0$)			
Pattern	Label	sh_1	sh_2
I	N vs N	0	0
IV	NM vs NM ($sh_1 = sh_2$)	3	3
IV	NM vs NM ($sh_1 = sh_2$)	6	6
H_1 patterns ($\delta = 1$)			
I	N vs N	0	0
II	N vs NM	0	6
III	NM vs N	6	0
IV	NM vs NM ($sh_1 = sh_2$)	6	6
V	NM vs NM ($sh_1 < sh_2$)	3	6
VI	NM vs NM ($sh_1 > sh_2$)	6	3

Table 2. The six selected patterns under null (H_0) and alternative (H_1) hypotheses. I. two normals; II. normal vs mixture (sh = 6); III. mixture (sh = 6) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2 = 6$); V. two mixtures with different shifts ($sh_1 = 3 < sh_2 = 6$); VI. two mixtures with different shifts ($sh_1 = 6 > sh_2 = 3$). Abbreviations: N = Normal distribution; NM = Mixture of Normal distributions.

As regards the nonparametric tests, exact p-values have been computed for the KS test, while for the WMW, PG2, Cucconi, CvM and AD tests we report the asymptotic p-values. For the CvM test we used the p-values tabulated by Anderson et al. ([25]), since it has been shown that under

H_0 the two-sample statistic has the same limiting distribution as that of the one-sample statistic ([33]). Moreover, using the Burr's formula, we can obtain the p-values corresponding to all the possible values of the CvM statistic and not limited to the tabulated ones; it is reported that such an approximation works to the fifth decimal place for values of the statistic between 0.42 and 2.2, and we empirically verified that, for values of the statistic greater than 0.10308, the formula approximates up to the second decimal place the p-values tabulated by Anderson et al. ([25]). Moreover, values of the statistic below 0.10308 correspond to p-values higher than 0.57, which are of no interest here since they indicate non significant features. Finally, for the three Zhang tests, as well as for the Cucconi test as an alternative to the asymptotic version, we used the Monte Carlo approach to find the corresponding approximate empirical p-values (size = 2000 simulations). The tests were applied with no correction for ties since the considered data are continuous and do not have tied values. The *rnormmix* function from the *mixtools* package was used to simulate the mixtures of univariate normal distributions. Because of the computational burden of the simulation, parallel programming (using the two packages *doParallel* and *doRNG*) was implemented in order to perform simultaneous and reproducible computations.

2.4.2 Simulation results

A complete report of the simulation results is shown in the Appendix 3. Given 10000 simulations, the chosen 5% significance level provided a SE equal to 0.2%, which is the simulation error for the H_0 patterns; as regards the power, in the worst situation when it is equal to 50%, the SE would be equal to 0.5%. Therefore, we got a precision of simulation estimates up to the third decimal. All the tests resulted robust according to both Conover and Marozzi indications, since relative frequencies of H_0 rejection under H_0 (patterns I and IV) were less than 0.1 and 0.075 respectively; the maximum frequency was 0.063 for the Z_k test, when $\lambda = 0.95$ and $(m,n) = (20,20)$; the KS test was the only one with frequencies lower than 0.05 in most situations, reaching a minimum of 0.034, when $\lambda = 0.80$ and $(m,n) = (20,20)$. As regards the power, we did not find a clear winner for all the patterns; in general, within the same category (location, or location-scale, or general two-sample problem) the tests shared similar power for all the considered patterns, except for KS and Z_k which showed to be very conservative tests. Moreover, as expected, the tests for the general two-sample problem were generally more sensitive than those for the location-scale problem, especially when $\lambda = 0.95$. As an example to visualize the overall advantage brought by the general two-sample problem tests, we report the results in terms of power under the patterns I-VI, having fixed $\delta = 1$ and $m = 20$ vs $n = 20$ and with $\lambda = 0.80$ (Figure 2) and $\lambda = 0.95$ (Table 3).

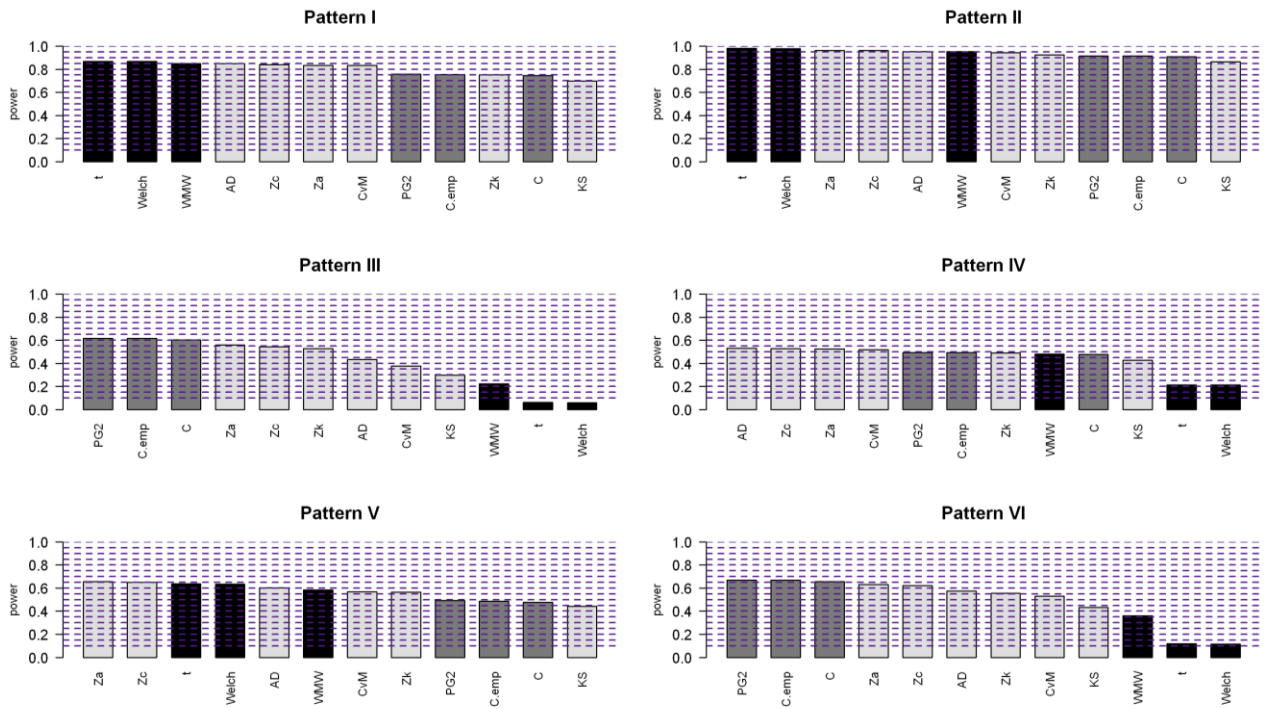


Figure 2. Barplots of power of the considered two-sample tests for the six selected scenarios with $\delta=1$, $\lambda=0.80$ and $(m,n) = (20,20)$. I. two normals; II. normal vs mixture ($sh=6$); III. mixture ($sh=6$) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2=6$); V. two mixtures with different shifts ($sh_1=3 < sh_2=6$); VI. two mixtures with different shifts ($sh_1=6 > sh_2=3$). All mixtures have $\lambda = 0.80$. The tests are sorted in descending order according to the power. The different colors indicate the three types of tests (location tests in black, tests for the location-scale problem in gray and tests for the general two-sample problem in lightgray).

Pattern	Tests											
	Student	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.869	0.868	0.849	0.757	0.745	0.755	0.698	0.834	0.848	0.752	0.839	0.834
II	0.904	0.901	0.883	0.804	0.794	0.803	0.745	0.869	0.880	0.799	0.879	0.877
III	0.419	0.417	0.697	0.638	0.623	0.636	0.589	0.720	0.723	0.619	0.690	0.680
IV	0.499	0.497	0.750	0.677	0.661	0.674	0.634	0.761	0.763	0.663	0.729	0.717
V	0.729	0.726	0.770	0.682	0.666	0.680	0.636	0.768	0.772	0.669	0.745	0.737
VI	0.485	0.483	0.743	0.680	0.663	0.678	0.634	0.761	0.764	0.663	0.733	0.722

Table 3. Power estimates with $\alpha=0.05$, $\delta=1$, $\lambda=0.95$ and $(m,n) = (20,20)$. Location tests: t = Student's t test; Welch = Welch's t test; WMW = Wilcoxon-Mann-Whitney test. Location-scale tests: PG2 = Podgor-Gastwirth PG2 test; C = Cucconi test (asymptotic version); C.emp = Cucconi test (empirical version). Tests for the general two-sample problem: KS = Kolmogorov-Smirnov test; CvM = Cramer-von Mises test; AD = Anderson-Darling test; Z_K = Zhang Z_K test; Z_C = Zhang Z_C test; Z_A = Zhang Z_A test.

With both mixture weights, the two parametric location tests (Student's t and Welch's t) headed the power ranking in case of two normal distributions (pattern I) or when the second distribution was a mixture (pattern II); in the latter case their ability lied in detecting the observations in the tail of the mixture. However, when the two ECDFs were crossing, *i.e.* when the two distributions overlapped at certain points, their power collapsed (see Figures A2 in the Appendix 2, patterns III, IV, VI). The location tests (Student's t, Welch's t and WMW) generally showed a high power for pattern V, where the two ECDFs of the mixtures appeared mostly separated, and thus the differences between the two samples were mainly in terms of location. The nonparametric location test, *i.e.* the WMW, was more powerful than the parametric tests in the patterns involving two mixture distributions, except for pattern V and $\lambda = 0.80$ (Tables A3.1.6,

A3.1.7). However, it did not emerge as the best alternative to the parametric tests in presence of two mixture distributions, especially when $\lambda = 0.80$, where the advantage of the location-scale and general two-sample tests was more evident. The location-scale tests (PG2 and Cucconi tests) showed the highest power in the patterns III and VI with $\lambda = 0.80$ (Tables A3.1.6, A3.1.8), corresponding to situations of scale differences being one distribution in the middle of the other one with ECDFs overlapping for the most part. Such tests seem to be particularly able to detect the differences in the peaks of the compared distributions. In general, the PG2 test was more powerful than the Cucconi test (both asymptotic and empirical versions) and the approximate empirical version of the Cucconi test was always more powerful than the asymptotic version. In the patterns IV and V with $\lambda = 0.80$ and the patterns III-VI with $\lambda = 0.95$ the tests for the general two-sample problem were generally the most powerful ones. The most liberal tests were the AD test, its analog Z_A test, the CvM test and its analog Z_C test; in particular, for mixtures with equal shifts (pattern IV) the AD test was the most sensitive one, together with the Z_C and CvM tests. Moreover, when the normality assumption was fulfilled, the AD, CvM, Z_A and Z_C tests had a limited loss in power compared to that of the other nonparametric tests, while being very powerful in detecting any difference between the two samples in the remaining patterns. For example, the application of the AD test implied a loss in power of $\sim 2\%$ in pattern I, but a gain of $\sim 32\%$ in pattern IV respect to the parametric tests (Table A3.1.6). It is worth to notice that, in spite of being tests for the general two-sample problem, the KS and its analog Z_K proved to be very conservative, showing low power in all the simulated patterns. With small weights to the first component of the mixtures ($\lambda = 0.20$ and $\lambda = 0.05$), we obtained similar results, being the AD, CvM, Z_A and Z_C the most sensitive tests in the III-VI patterns, while still maintaining a high power in the I and II patterns. Respect to $\lambda = 0.80$ and $\lambda = 0.95$, the power was higher for all the tests and, for patterns II, III and V, it often reached the 100%. Indeed, in these cases the mixture distribution density is concentrated at its second component, thus yielding well separated distributions and easily detectable differences. Regarding the unbalanced sample size settings ($m=20$ vs $n=40$ and $m=40$ vs $n=20$), the dominance of the general two-sample tests remained evident, except for the pattern V with $m=20$ vs $n=40$ and $\lambda = 0.80$, where the Welch's t test resulted as the most powerful test, even if the difference in power was small ($\sim 4\%$) respect to the Z_A test (Table A3.1.7). An explanation could be the presence of a large tail of the distribution of the second sample, corresponding to an evident difference between the two means (Figure A2.1, $m=20$ vs $n=40$, pattern V).

2.4.3 Real data example

As an example of the application of the considered tests to real data the dataset included in the R Bioconductor package *breastCancerNKI* was chosen, containing gene expression data as

published in Van't Veer et al. ([34]) and Van de Vijver et al. ([35]) (24481 genes/features evaluated in 337 samples). We defined two classes according to the Estrogen Receptor (ER) status (249 ER positive and 88 ER negative). To prevent confounding, we matched 1:1 the subjects on the basis of three clinical variables, *i.e.* age (quintile based classes), tumor size (quintile based classes) and tumor grade (3 classes). Thus, we obtained 33 ER positive paired with 33 ER negative individuals, a sample size setting in between those considered for the simulation study. We then used a 100% filtering (*i.e.* we required the detection of all features in all 66 samples), keeping the expression of 19264 genes. We then applied all the tests, adjusting for multiple testing with the Benjamini-Hochberg method ([36]) for the control of the False Discovery Rate (FDR). We considered a test as significant when the corresponding FDR-adjusted p-value was $< 5\%$. Globally, the AD test resulted as the most 'saving-features' test, with 3512 detected genes; then, in descending order according to the number of significant features, we got the CvM, WMW, Z_k , Z_A , Student's t, Cucconi (empirical version), Welch's t, PG2, Z_C , Cucconi (asymptotic version) and the KS tests (**Table 4**). Over the 19264 filtered genes, 15094 were not significantly DE and 1842 were significantly DE for all tests. Focusing on the remaining subset of 2328 'gray-zone' features and classifying the tests according to the three categories (location, location-scale and general two-sample problem), the largest number of DE features was detected by the WMW test for the location problem (1491 features), the Cucconi (empirical version) for the location-scale problem (1051 features) and the AD test for the general two-sample problem (1670 features). The Cucconi test detected 31 features more than the PG2 test; however, such a slight difference could be dependent on the choice of the real data. The application of the nonparametric tests added 151 significantly DE features respect to the parametric tests. For illustration purposes the distribution of a representative feature among such 151 is represented in **Figure 3A** according to the ER status; as expected, in both ER classes the distributions appeared not normal (skewed and multimodal). The KS and its analog Z_k acted differently respect to the other tests as previously seen in the simulation, since 131 features were detected using all tests except KS (representative feature: **Figure 3B**) and 84 features using all tests except Z_k (representative feature: **Figure 3C**). Thirty-seven features were separately recognized by the location-scale tests (PG2 and Cucconi), being characterized by overlapped values in the two classes but with different peak magnitudes (representative feature: **Figure 3D**).

In **Figure 4** we plotted a Venn diagram considering the three most sensitive tests within each category according to the simulation results, *i.e.* the WMW test (location), the PG2 (location-scale problem) and the AD test (general two-sample problem).

Excluding the 781 features in common among the three tests, the AD test shared 674 features with the WMW test and 102 features with the PG2, while WMW and PG2 tests shared no features. Similar results were obtained using the Cucconi test (both versions) instead of PG2.

Test	Test category	N° of significantly DE features
Anderson-Darling	3	3512
Cramér-von Mises	3	3422
Wilcoxon-Mann-Whitney	1	3333
Zhang Z_K	3	3323
Zhang Z_A	3	3301
Student's t test	1	2894
Cucconi (empirical)	2	2893
Podgor-Gastwirth PG2	2	2862
Welch's t test	1	2856
Zhang Z_C	3	2757
Cucconi (asymptotic)	2	2565
Kolmogorov-Smirnov	3	2535

Table 4. Real data example: ranking of the tests according to the number of significantly DE features (FDR-adjusted p-value < 0:05).

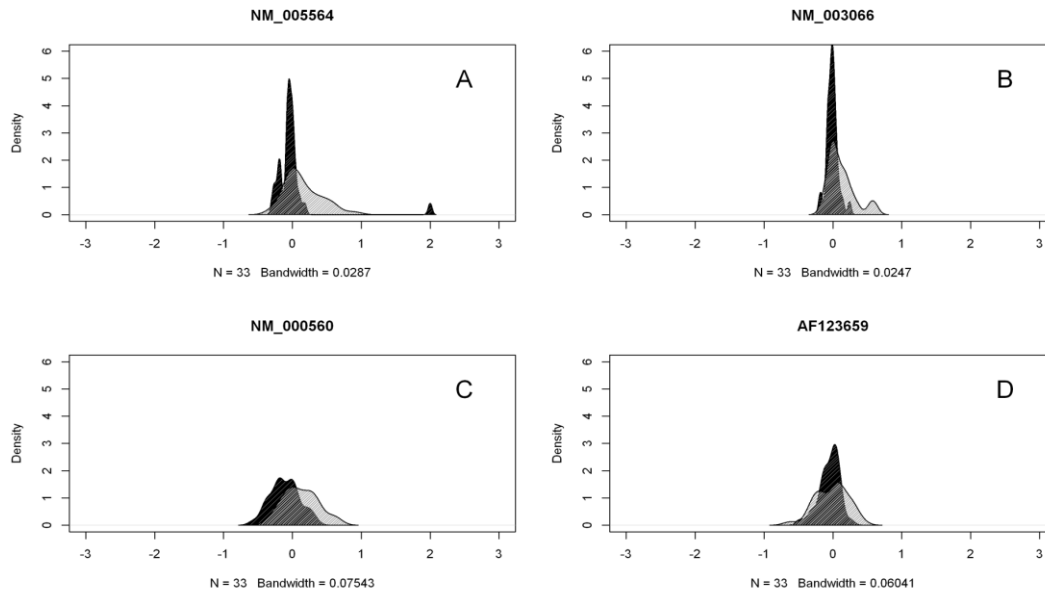


Figure 3. Real data example: exemplificative kernel density distributions of 4 selected 'gray-zone' genes in ER positive (in black) and ER negative (in gray) subjects.

Thus, the tests for the location-scale problem seemed to be less concordant with the WMW location test. On the contrary, the AD test and the WMW test shared in all 1455 (63%) of the features, being the AD a more sensitive test. As expected, regarding the features separately identified by each of the three tests, we noticed that the 36 features detected by the WMW test were almost symmetric distributions and mainly differed in terms of location (**Figure A4.1**), 137 features had different peak magnitudes mainly differing in terms of scale, and thus were significant at PG2 test only (**Figure A4.2**), while the distributions of the 113 features detected as DE by the AD test were different in general aspects, *i.e.* besides location and scale, they showed different and irregular tails (**Figure A4.3**).

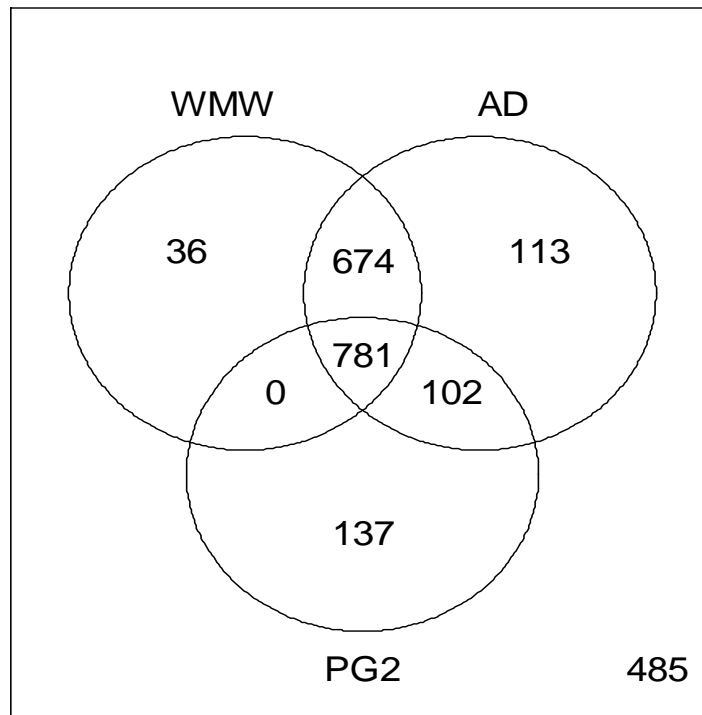


Figure 4. Real data example: Venn diagram with WMW, PG2 and AD tests on the 'gray-zone' subset of features.

2.4.4 Final remarks

Two-sample tests for class comparison are often used in bioinformatics and medicine for exploratory purposes, *i.e.* to detect DE features between different biomedical conditions. The most commonly used are the location tests, such as the parametric t test and its variations, together with nonparametric tests such as the WMW test; however, they are able to detect only shifts in distributions and not to identify any other difference in scale or shape. The major drawback in applying the above tests is that the expression values of high dimensional data generally exhibit departures from normality and features could be DE in other aspects rather than location only. Indeed, they can miss features which are characterized by more general and subtle distributional discrepancies. These different signals might hide differential biological processes and have to be preserved in order to be further explored by including them in an multivariable model for class prediction. We set a simulation study to evaluate the performance of different location tests (Student's t, Welch's t, WMW), tests for the location-scale problem (PG2 and Cucconi) and tests for the general two-sample problem (KS, CvM, AD, Z_K , Z_C , Z_A), by modeling the irregular signals by means of mixtures of two normal distributions. Although Z_C in particular was suggested as the best one among the three Z_K , Z_C and Z_A , we assessed the performance of all Zhang tests, since we wanted a complete comparison with the corresponding traditional tests (*i.e.* KS, CvM and AD). We did not find a clear winner for all considered distributional patterns among the tests proposed as an alternative to the most used ones.

However, the simulation study and the real data example showed that the tests for the general two-sample problem tend to save a great number of DE features, with a gain in power respect to the location and location-scale tests. Location tests consider DE a feature with almost symmetric distributions in the two compared samples, while location-scale tests are able to detect also differences in terms of peak magnitude. The tests for the general two-sample problem make one more step further introducing a more general concept of 'differential expression', thus overcoming the limitations of the above mentioned tests restricted to specific moments of the feature distributions. Specifically, the AD, CvM and their analogs Z_A and Z_C tests have to be preferred since their power was very similar to that of the more efficient parametric tests when the normality assumption was fulfilled, while in all the other situations they still resulted very powerful in detecting differences between the two samples. The AD test in particular seemed to be a very sensitive test in most of the simulated patterns and, accordingly, kept more features than any other considered test in the real data example. In conclusion, the AD test should be considered as a powerful alternative to the parametric tests as feature screening method in order to keep as many discriminative features as possible for the subsequent class prediction analysis.

2.5 Combined use of t and Anderson-Darling tests

Given the final remarks about the results of the simulation study (see paragraph 2.4), we propose the combined use of t and AD tests in order to perform the class comparison analysis between two classes, *i.e.* hemolyzed samples (cases) vs not hemolyzed samples (controls).

The t test is commonly applied for class comparison being directly related to the fold change (FC), which gives a measure of the direction and the strength of the differential expression; however, this aspect could represent a limitation because only the difference between the means is explored. On the other hand, the AD test is able to detect more general differences between the two classes, which could reveal hidden differential biological processes.

We report below an example of volcano and concordance plots (**Figure 5**). We considered DE the features for which the p-value was below the 5% level for at least one of the two tests. This procedure could inflate the overall Type I error; however, we expect such an effect to be marginal because the two tests statistics are likely to be dependent and, in addition, both the tests are applied to the same data.

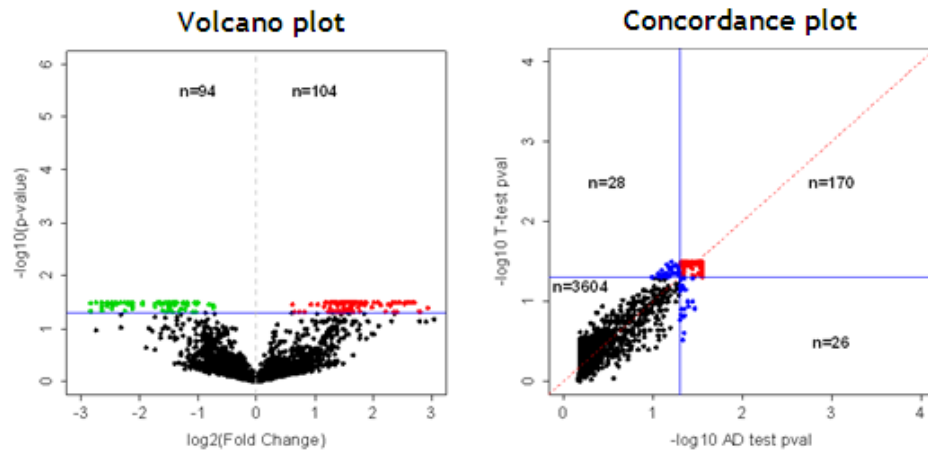


Figure 5. Class comparison results: t-test volcano plots and concordance plots between t- and Anderson and Darling test. In the volcano plots the \log_2 feature fold change is plotted on the x-axis and the negative \log_{10} p-value at t-test is plotted on the y-axis. The horizontal line indicates the 5% significance level, while n is the number of significantly up-regulated (first quadrant) and down-regulated (second quadrant) features. In the concordance plots the negative \log_{10} p-value according to the AD test is plotted on the x-axis and the negative \log_{10} p-value according to the t-test is plotted on the y-axis. Points lying on the dashed line would indicate perfect concordance between the two tests.

Looking at the volcano plot, 104 features resulted as significantly up-regulated and 94 features as significantly down-regulated with the t test, for a total of 198 features. The concordance plot shows that 224 features were significantly at the t or AD test: 170 features were detected by both the tests (first quadrant of the plot in the right panel), 28 features only by the t test (second quadrant) and 26 features only by the AD test (fourth quadrant).

3. Class prediction analysis

In class prediction analysis, as for the class comparison analysis, the comparison groups are predefined but the aim is to develop a classifier, *i.e.* a feature expression-based multivariable model able to accurately discriminate between cases and controls.

3.1 First step: feature selection

Repeated bootstrap samples ($B=1000$) [40] were drawn from the original dataset in order to obtain a robust ranking of features with different distribution between two classes of interest, on the basis of their frequency of simultaneous extraction by three machine learning selection algorithms on all the bootstrap samples, *i.e.* Prediction Analysis for Microarrays (PAM) [41], Random Forests (RF) with Boruta FS (feature selection) method [42] and the Elastic SCAD SVM [43] or, alternatively, the L_1 -SVM [44]. These methods were chosen since they overcome the ‘curse of dimensionality’ usually present in high dimensional data and because the inherent biological dependence between the features implies correlation between features. In fact, both PAM and RF outperform standard SVM in the presence of correlated predictors [45], and the same is true for the Elastic SCAD SVM, which is a SVM variant. On the other hand, the L_1 -SVM is computationally faster than the Elastic SCAD SVM and tends to select only one feature from a group of correlated features, dropping the others. To guarantee the reproducibility of the results, the *bootfs* package is used, by properly modifying the *doBS* function. The change consisted in the introduction of a seed for the bootstrap sampling, in addition to the seeds for the classification algorithms.

3.1.1 Prediction Analysis for Microarrays

PAM, also called the nearest shrunken centroids, is an enhancement of the simple nearest centroid classifier [41]. Such a technique can be seen as a modification to the classic Linear Discriminant Analysis (LDA) in two aspects tailored to high dimensional and low-sample sized data, *i.e.* the regularization of the covariance matrix and the variable selection through shrinkage. Briefly, the method computes a standardized centroid for each class. This is the average expression for each feature in each class divided by the within-class standard deviation for that feature. Nearest centroid classification takes the feature expression profile of a new sample, and compares it to each of these class centroids. The class whose centroid that it is closest to, in squared distance, is the predicted class for that new sample. Nearest shrunken centroid classification makes one important modification to standard nearest centroid classification. It ‘shrinks’ each of the class centroids toward the overall centroid for all classes

by an amount, *i.e.* the threshold. This shrinkage consists of moving the centroid towards zero by threshold and setting it equal to zero if it hits zero. For example, if threshold is 2.0, a centroid of 3.2 is shrunk to 1.2, a centroid of -3.4 is shrunk to -1.4, and a centroid of 1.2 is shrunk to zero. After shrinking the centroids, the new sample is classified by the usual nearest centroid rule, but using the shrunk class centroids. More formally, let x_{ij} be the expression for features $i = 1, 2, \dots, p$ and samples $j = 1, 2, \dots, n$. Given the classes $1, 2, \dots, K$, let C_k be the indices of the n_k samples in class k . The i^{th} component of the centroid for class k is

$$\bar{x}_{ik} = \frac{\sum_{j \in C_k} x_{ij}}{n_k}$$

the mean expression value in class k for feature i ; the i^{th} component of the overall centroid is

$$\bar{x}_i = \frac{\sum_{j=1}^n x_{ij}}{n}$$

So, the class centroids is shrunk toward the overall centroids after standardizing by the within-class standard deviation for each feature. This standardization has the effect of giving higher weight to features whose expression is stable within samples of the same class. Such standardization is inherent in other common statistical methods such as LDA.

Let

$$d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k(s_i + s_0)} \quad (1)$$

where s_i is the pooled within-class standard deviation for feature i :

$$s_i = \sqrt{\frac{1}{n - K} \sum_k \sum_{j \in C_k} (x_{ij} - \bar{x}_{ik})^2} \quad (2)$$

and

$$m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}}$$

makes the $m_k * s_i$ equal to the estimated standard error of the numerator in d_{ik} . In the denominator, the value s_0 is a positive constant (with the same value for all features), included to guard against the possibility of large d_{ik} values arising by chance from features with low expression levels. s_0 is set equal to the median value of the s_i over the set of features. A similar strategy is used in the SAM methodology [46]. Thus d_{ik} is a t statistic for feature i , comparing class k to the overall centroid. Equation (1) can be rewritten as

$$\bar{x}_{ik} = \bar{x}_i + m_k(s_i + s_0)d_{ik} \quad (3)$$

PAM shrinks each d_{ik} toward zero, giving d_{ik}' and yielding shrunken centroids

$$\bar{x}_{ik}' = \bar{x}_i + m_k(s_i + s_0)d_{ik}' \quad (4)$$

The shrinkage used is called soft thresholding, *i.e.* each d_{ik} is reduced by an amount Δ in absolute value and is set to zero if its absolute value is less than zero. Algebraically, soft thresholding is defined by

$$d_{ik}' = \text{sign}(d_{ik})(|d_{ik}| - \Delta)_+ \quad (5)$$

where $_+$ means positive part ($t_+ = t$ if $t > 0$ and zero otherwise). Because many of the \bar{x}_{ik} values will be noisy and close to the overall mean \bar{x}_i , soft thresholding usually produces more reliable estimates of the true means [47; 4848]. The shrinkage has two advantages: 1) it can make the classifier more accurate by reducing the effect of noisy features, 2) it does automatic FS. In particular, if a feature is shrunk to zero for all classes, then it is eliminated from the prediction rule. Alternatively, it may be set to zero for all classes except one, and therefore it means that high or low expression for that feature characterizes that class. The practitioner has to set the value to use for threshold. To guide in this choice, PAM does k-fold Cross Validation (CV) for a range of threshold values. The samples are divided up at random into k roughly equally sized parts. For each part in turn, the classifier is built on the other k-1 parts then tested on the remaining part. This is done for a range of threshold values and the CV misclassification error rate is reported for each threshold value; the threshold value which gives the minimum CV misclassification error rate is the chosen one. At the end a simple-to-understand classifier is obtained, but there are potential limitations with such a method. The main concern is that PAM is possibly too extreme, since the covariance matrix is restricted to be diagonal. It may be beneficial to differently estimate the covariance matrix; furthermore, more effective shrinkage schemes may be possible.

3.1.2 Random Forest

Bagging or bootstrap aggregation is a technique for reducing the variance of an estimated prediction function. The essential idea of bagging is to apply an estimator to multiple bootstrap samples and averaging the result across bootstrap samples, thus reducing the variance. Decision trees [4949] are ideal candidates for bagging, since they can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias. Since trees are notoriously noisy, they benefit greatly from the averaging. Moreover, the expectation of an average of B trees is the same as the expectation of any of them. Therefore, the bias of bagged trees is the same as that of the individual trees, and the only possibility of improvement is through variance reduction. For classification problems, a group of trees is finally constructed,

each tree casting a vote for the predicted class. RF is a substantial modification of bagging which builds a large collection of de-correlated trees and then averages them. It is among the most popular machine learning methods and it is often applied in the life sciences because it supports high dimensional datasets, is robust to large amounts of noise, requires little parameter tuning and demands no predictor transformation [50; 51; 52; 53]. RF also natively produces a feature importance measure that directly expresses the role of a feature in all interactions utilized in the model, including weak and multivariate ones. These characteristics make RF a promising classification algorithm for FS tasks [51]. RF is designed to form an ensemble of weak unbiased classification trees. Each tree is constructed using different bootstrap samples. Each bootstrap sample is a result of drawing with replacement the same number of objects as in the original training set. As a result, a part of objects is not used for building a tree but it is used for performing an Out Of Bag (OOB) error estimate and for importance measurement. At each step of the tree construction a different subset of features is randomly selected. The split is performed using the attribute which leads to the best distribution of data between nodes of the tree. This procedure is performed until the whole tree is built. Then the constructed tree is used to classify its OOB objects, and the result is used for obtaining the approximations of the classification error and for the computation of confusion matrices. New objects are classified by all trees in the forest and the final decision is made by simple voting (see below the Random Forest Algorithm). The importance of each feature is estimated as follows. Firstly, the classification of all objects is performed. The number of votes for a correct class is recorded for each tree. Then the values of a given feature are randomly permuted across objects, and the classification is repeated. The number of votes for a correct class is again recorded for each tree. The importance of the feature for the single tree can be then defined as a difference between the number of correct votes for the original and permuted system divided by the number of objects. The importance of the feature is then obtained by averaging importance measures for individual trees. The Z-score, *i.e.* the ratio between the average value and its standard deviation, can also be used as an importance measure. The advantage of the Z-score is that it gives more weight to a relatively small but stable decrease of classification performance. RF provide two straightforward measures of feature importance, *i.e.* Mean Decrease Impurity (MDI) and Mean Decrease Accuracy (MDA) [54]. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. When training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

Random Forest Algorithm

1. For $b=1$ to B :
 - (a) Draw a bootstrap sample Z^* of size N from the training data.
 - (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{mn} is reached:
 - i. select m features at random from the p features;
 - ii. pick the best feature/split-point among the m ;
 - iii. split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x :

Regression:

$$\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$$

Classification: Let $\hat{C}_b(x)$ be the class prediction of the b^{th} RF tree. Then

$$\hat{C}_{rf}^B(x) = \text{majority vote} \left\{ C_b(x) \right\}_1^B$$

In formulas, the importance of a feature X_m for predicting Y by adding up the weighted impurity decreases $p(t)\Delta_i(s_t, t)$ for all nodes t where X_m is used, averaged over all N_T trees in the forest:

$$\text{Imp}(X_m) = \frac{1}{N_T} \sum_T \sum_{t \in T: v(s_t) = X_m} p(t) \Delta_i(s_t, t) \quad (6)$$

and where $p(t)$ is the proportion N_t/N of samples reaching t and $v(s_t)$ is the feature used in split s_t . When using the Gini index as impurity function, this measure is known as the Gini importance or Mean Decrease Gini. However, Equation (6) can be defined for any impurity measure $i(t)$. As mentioned before, a tree is made up of a hierarchy of nodes. A node takes incoming observations and sends them to its left or right child node according to a split rule. In so doing, a node attempts to purify the incoming dataset. A perfectly pure dataset is one that only contains one class of observations. Gini Impurity (7) is a measure of the deviation of a dataset from being perfectly pure.

$$I(\text{dataset}) = \sum_{i=0}^{\text{numClasses}} (p_i)(1 - p_i) \quad (7)$$

where p_i = proportion of class i in the dataset. Decrease in Gini Impurity (8) measures the change in impurity at a node caused by sending incoming observations left or right. A high Decrease in

Gini Impurity implies that the node and its corresponding feature were useful at separating the classes of interest.

$$\Delta I(Node) = I(d_{incoming}) - P_L I(d_L) - P_R I(d_R) \quad (8)$$

where $d_{incoming}$ = incoming dataset, $d_{L,R}$ = dataset sent left or right, n = number of observations in incoming dataset, $n_{L,R}$ = number of observations in left or right dataset, $P_L = n_L / n$ and $P_R = n_R / n$. Mean Decrease in Gini Impurity (MDG) averages the change in impurity seen across all nodes that used the same feature. This metric measures the forest-wide contribution of the feature at separating the different classes and constitutes one measure of feature importance.

However, FS based on impurity reduction is biased towards preferring features with more categories. Secondly, when the dataset has two (or more) correlated features, then any of these correlated features can be used as the predictor, with no concrete preference of one over the others. But once one of them is used, the importance of others is significantly reduced, since effectively the impurity they can remove is already removed by the first feature. As a consequence, they will have a lower reported importance. This is not an issue when one wants to use FS to reduce data overfitting, since it makes sense to remove features that are mostly duplicated by other features. But when interpreting the data, it can lead to the incorrect conclusion that one of the features is a strong predictor while the others in the same group are non-important, while actually they are very close in terms of their relationship with the response variable. The effect of this phenomenon is somewhat reduced thanks to random selection of features at each node creation, but in general the effect is not removed completely. As regards MDA, the classification results (predicted class vs actual class) of observations of the validation set form a tree confusion matrix, from which accuracy is calculated. Accuracy, which serves as an unbiased estimate of a tree performance, is used as a measure for both determining whether or not to discard a tree, and to calculate MDA. The MDA of a feature is the average change in accuracy across the forest, resulting from permuting the feature values in observations of the validation set. The classification results from each feature permuted observations form a new confusion matrix for each tree. Permuted accuracy for each feature is calculated from its corresponding confusion matrix. The change in accuracy (without permutation minus with permutation) is then calculated for each feature, for each tree, and averaged across the forest. This average (MDA) measures the forest-wide contribution of the feature at predicting the different classes. If permutation has no effect on accuracy (*i.e.* a MDA close to 0), one can conclude that the feature is non-important. If that permutation does have an effect, the change in accuracy is interpreted as the importance of that feature. Strobl et al. [55] compared both MDI and MDA and showed experimentally that the former is biased towards some predictor variables. As explained by White and Liu [56], in case of single decision trees this

bias stems from an unfair advantage given by the usual impurity functions $i(t)$ towards predictors with a large number of values. Strobl et al. [57] later showed that MDA is biased as well, and that it overestimates the importance of correlated features. From a theoretical point of view, Ishwaran [58] provides a detailed theoretical development of a simplified version of MDA, giving key insights for understanding the actual MDA.

RF are characterized by good accuracy, robustness and ease of use: in general they require very little feature engineering and parameter tuning. However, one drawback of this algorithm is the difficulty of interpreting the importance ranking of correlated features, in which case strong features can end up with low scores and the method can be biased towards features with many categories. Thus, it is clear that the importance score alone is not sufficient to identify meaningful correlations between features and the decision attribute. Breiman assumed that, due to low correlations between individual trees, the importance has normal distribution and hence Z-score can be used to assess importance of the variance. Unfortunately, it has been shown that Breiman assumption is false and therefore one needs some reference which can help to discern the truly important features from the non-important ones.

Regarding FS, such a process is intended to reduce input data dimension, by solving two important problems: facilitate learning accurate classifiers and discover the most interesting features, which may provide for better understanding of the problem itself [59]. Regarding the first problem, the objective is to learn an optimal classifier using a minimal number of features (minimal-optimal problem). Unfortunately, minimal-optimal is in general intractable even asymptotically, since there exist data distributions for which every feature subset must be tested to guarantee optimality [60]. Therefore it is common to resort to suboptimal methods. The second problem is motivated by recent applications in bioinformatics aimed at finding all features relevant to the target variable [61]. This defines the all-relevant problem. Thus, there are basically two classes of RF-based FS methods: all-relevant FS approaches (e.g. Artificial Contrasts with Ensembles (RF-ACE) [62] and Boruta [61]) and minimal-optimal approaches (e.g. Recursive Feature Elimination (RFE) and Regularized Random Forest (RRF) [62]). All-relevant FS approaches generally select a much larger number of features respect to RFE or RRF. Both the RF-ACE and Boruta algorithms are based on the idea first introduced in [63], *i.e.* the extension of the information system with shadows, which are artificial features created by permuting the order of values in the original data, and then the use of shadow importance scores to judge the score significance obtained by the actual features. However, the algorithms differ in the testing scheme used. RF-ACE performs a predefined number of iterations and, at each step, collects the importance of real features and the mean importance of all shadows. For each feature, a Student's t-test is applied to check whether its mean importance is significantly larger than the mean importance of the shadow attributes and features with p-values less than the chosen significance level α are returned as relevant. On the other hand, Boruta checks which features in

an iteration achieved higher importance than the best shadow; these events are counted for each feature until their number becomes either significantly higher or lower than what is expected at random, using a p-value cutoff. If the feature is deemed irrelevant, it is removed along with its shadow from the information system. This procedure is repeated until the status of all features is decided or until a previously set limit of iterations is exhausted; in the latter case, the status of some features may be undecided. Both RF-ACE and Boruta re-shuffle shadow features after every iteration. RFE is a group of methods where selection is performed by iterative stripping of less important features from the set until the classifier error becomes minimal. There are many implementations of this method that differ in the importance source used, the stripping criterion and the accuracy assessment method.

RRF is a modification of a RF that incorporates regularization into the tree growing algorithm [64; 65]. Specifically, RRF establishes a penalty for the use of a feature that is not previously used in a current tree construction. Such a penalty is proportional to the potential information gain from building a split on this feature, so that the selected features are only the ones with significant information that is not redundant with respect to already built splits, therefore only a subset of all features is actually used in the ensemble.

The Boruta Algorithm

Boruta algorithm, being an all-relevant FS method, tries to find all features carrying information usable for prediction, rather than finding a possibly compact subset of features on which some classifier has a minimal error. It derives its name from a Slavic spirit of the forest, since the first version of Boruta was a wrapper over the RF method. The principle of the algorithm is in line with the RF theory, *i.e.* it copes with problems by adding more randomness to the system: a randomized copy of the system is created and merged with the original and a classifier is built for this extended system. To assess the importance of a feature in the original system, this feature is compared with that of the randomized features. Only features for whose importance is higher than that of the randomized features are considered important. Specifically, the following procedure is applied:

- an extended system is built, where each feature is replicated. The values of replicated features are then randomly permuted across the objects; as a consequence, all correlations between the replicated features and the decision feature are random by design;
- several RF runs are performed, the replicated features are randomized before each run, and therefore the random part of the system is different for each RF run;

- for each run the importance of all features is computed;
- a feature is deemed important for a single run if its importance is higher than maximal importance of all randomized features;
- a statistical test for all features is performed. The null hypothesis is that importance of the feature is equal to the Maximal Importance of the Random Attributes (MIRA). The test is a two-sided equality test; the hypothesis may be rejected either when importance of the feature is significantly higher or significantly lower than MIRA. For each feature it is counted how many times the importance of the feature is higher than MIRA (a hit is recorded). The expected number of hits for N runs is $E(N) = 0.5N$ with standard deviation $S = \sqrt{0.25N}$ (binomial distribution with $p = q = 0.5$). A feature is deemed important (accepted) when the number of hits is significantly higher than the expected value and is deemed non-important (rejected), when the number of hits is significantly lower than the expected value. It is straightforward to compute limits for accepting and rejecting features for any number of iterations at a specified confidence level α ;
- features which are deemed non-important are removed from the information system, usually with their randomized mirror pair. In some cases the randomized features can be kept in the system - it may help in reduction of the number of features deemed important, without reducing accuracy of the RF classifier;
- the procedure is performed for predefined number of iterations or until all features are either rejected or conclusively deemed important, whichever comes first. In the former case, there are features left, which are neither approved nor rejected, and are further referred to as undetermined.

3.1.3 Support Vector Machines

SVM are supervised machine learning models introduced in 1992 by Boser, Guyon, and Vapnik [66; 67]. The SVM classifier is widely used in bioinformatics due to its high accuracy, ability to deal with high dimensional data, and flexibility in modeling different sources of data [68]. SVM belong to the general category of kernel methods [69; 70]. A kernel method is an algorithm that depends on the data only through dot-products. The dot product can be replaced by a kernel function which computes a dot product in some possibly high dimensional feature space. This gives two advantages: first, the ability to generate nonlinear decision boundaries using methods designed for linear classifiers; second, the use of kernel functions allows the user to apply a classifier to data that have no obvious fixed-dimensional vector space representation. When

training a SVM, the practitioner needs to decide how to pre-process the data, what kernel to use, and finally, set the parameters of the SVM and the kernel. Specifically, SVM constructs a hyperplane or a set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression or other tasks; it aims at separating the classes with a decision surface that maximizes the distance between the classes (the functional margin), since in general the larger the margin the lower the generalization error of the classifier. The surface is often called the optimal hyperplane, and the data points closest to the hyperplane are called Support Vectors (SV) (**Figure 6**). Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space.

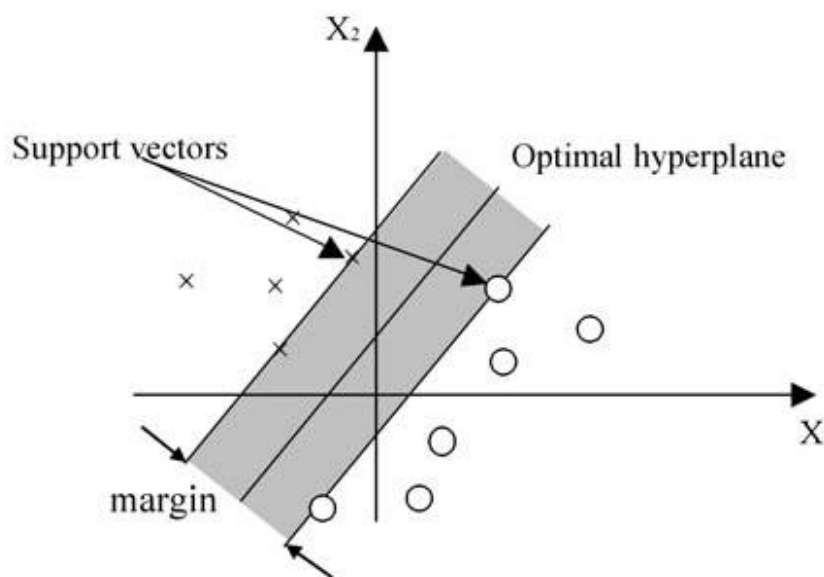


Figure 6. A SVM classification defined by a linear hyperplane that maximizes the separating margins between the classes.

Therefore, the original finite-dimensional space can be mapped into a much higher-dimensional space, presumably making the separation easier in that space (**Figure 7**) [71]. In order to keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot-products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x_i, x_j)$ selected to suit the problem. Often not only a prediction rule is needed but one also wants to identify relevant components of the classifier. Thus, it would be useful to combine FS methods with SVM classification, in order to find relevant features for prediction. In this context, the objective of FS is three-fold: (i) improving the prediction performance of the predictors, (ii) providing faster and more cost-effective predictors, (iii) gaining a deeper insight into the underlying processes that generated the data. There are three main groups of FS methods: filter, wrapper and embedded methods [72; 73; 74; 75; 76]. Filter methods simply rank individual features by independently assigning a score to

each feature. These methods ignore redundancy and fail in situations where only a combination of features is predictive.

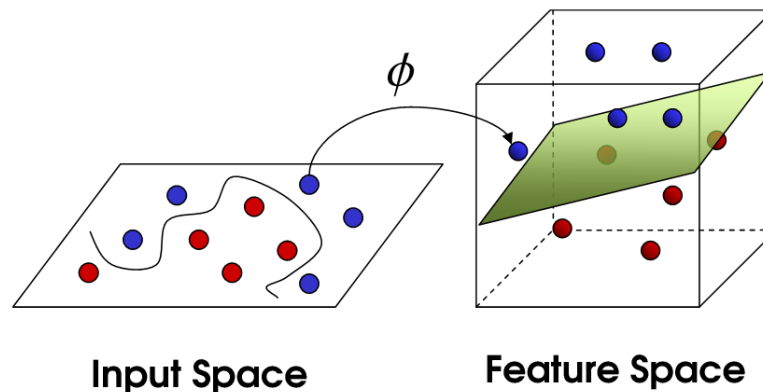


Figure 7. From Input Space to Feature Space.

Also, if there is a pre-set limit on the number of features to be chosen (e.g. top 10 features), this limit is arbitrary and may not include all informative features. Connecting filtering with a prediction procedure, wrapper methods wrap FS around a particular learning algorithm. Thereby, prediction performance of a given learning method assesses only the usefulness of subsets of features. After a subset with lowest prediction error is estimated, the final model with reduced number of features is built [75]. However, wrapper methods have the drawback of high computational burden, making them less applicable when the dimensionality increases. Wrapper methods also share the arbitrariness of filter methods in FS. The third group of FS procedures are embedded methods, which perform FS within learning classifiers to achieve better computational efficiency and better performance than wrapper methods. The embedded methods are less computationally expensive and less prone to data overfitting than the wrapper methods [59]. Among the latter, Guyon [72] proposed the RFE algorithm, which iteratively keeps a subset of features that are ranked by their contribution to the classifier. This approach is computationally expensive and selecting features based only on their ranks may not derive acceptable prediction rules. An alternative to SVM with RFE is to use penalized SVM with appropriate penalty functions. The penalty function controls the trade-off between allowing training errors and forcing rigid margins and creates a soft margin that permits some misclassifications, such as it allows some training points on the wrong side of the hyperplane. Increasing the penalty increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well. Penalized SVM belongs to the group of the embedded methods and provides automatic FS. There are several penalization functions such as Ridge, LASSO, SCAD, Elastic Net [48; 77; 78] and Elastic SCAD. The Ridge penalty [74] corresponds to the ordinary SVM, which does not provide any FS. Since SVM is extremely sensitive to the choice of tuning parameters, the search for optimal parameters becomes an

essential part of the classification algorithm [79]. More formally, suppose a training dataset with input data vector $x_i \in \mathbb{R}^p$ and corresponding class labels $y_i \in \{-1, 1\}$, $i = 1, \dots, n$ is given. The SVM finds a maximal margin hyperplane such that it maximizes the distance between classes. A linear hyperplane can always perfectly separate n samples in $n + 1$ dimensions. Assuming that data are linearly separable [76], a linear classifier is based on a linear discriminant function:

$$\{x : f(x) = \mathbf{w}^T \mathbf{x} + b = 0\}, \quad (9)$$

where $\mathbf{w} = (w_1, w_2, \dots, w_p)$ is the weight vector, *i.e.* a unique vector of coefficients of the hyperplane with $\|\mathbf{w}\|_2 = 1$, b is the intercept of the hyperplane and the symbol ‘ \cdot ’ denotes the inner product operator. The class assignment for a test data vector $x_{\text{test}} \in \mathbb{R}^p$ is given by $y_{\text{test}} = \text{sign}[f(x_{\text{test}})]$ (Figure 8). Soft margin SVM allows some data points to be on the wrong side of the margin. To account for erroneous decisions, slack variables $\xi_i \geq 0$, $i = 1, \dots, n$ are defined as the distance between a misclassified data point and the corresponding margin. For data points on the correct side of the margin $\xi_i = 0$, for data points inside the margin $0 < \xi_i \leq 1$ and for misclassified data points $\xi_i > 1$. The sum of non-zero ξ_i is penalized with a cost parameter C and then added to the optimization function penalty in the minimization problem:

$$\begin{aligned} \min_{b, \mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n \xi_i \\ \text{s. t.} \quad & \\ & \xi_i \geq 0 \\ & y_i(b + \mathbf{w}x_i) \geq 1 - \xi_i, \quad i = 1, \dots, n \end{aligned} \quad (10)$$

The optimization problem (10) is called the soft margin SVM. The cost parameter C is a data dependent tuning parameter that controls the balance between minimizing the coefficients of the hyperplane and correct classification of the training dataset. C is often chosen by CV. Problem (10) can be solved by using convex optimization techniques, *i.e.* the method of Lagrange multipliers [74]. Convex optimization techniques provide a unique solution for hyperplane parameters \mathbf{w} and b :

$$\hat{\mathbf{w}} = \sum_{i=1}^n \alpha_i y_i x_i \quad (11)$$

where $\alpha_i \geq 0$, $i = 1, \dots, n$ are Lagrange multipliers. The data points with positive α_i , are the SV. All data points lying on the correct side of their margin have $\alpha_i = 0$. Thus, they do not have any impact on the hyperplane, and Equation (11) can be rewritten as

$$\hat{\mathbf{w}} = \sum_{s \in S} \alpha_s y_s \mathbf{x}_s \quad (12)$$

where the set of indices of the SV S is determined by $S := \{i : \alpha_i > 0\}$. The coefficient \hat{b} can be calculated from $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i + \hat{b}) = 1 - \xi_i$ for any i with $\alpha_i > 0$.

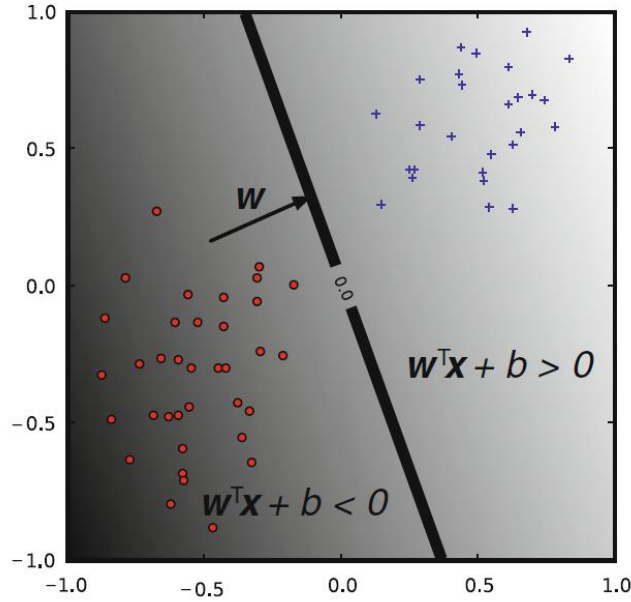


Figure 8. A linear classifier. The hyperplane (line in 2-d) is the classifier decision boundary. A point is classified according to which side of the hyperplane it falls on, which is determined by the sign of the discriminant function.

Usually, an average of all solutions for \hat{b} is used for numerical stability.

Regarding penalized SVM, Hastie et al. [74] showed that the SVM optimization problem is equivalent to a penalization problem which has the ‘loss and penalty’ form

$$\min_{b, \mathbf{w}} \frac{1}{n} \sum_{i=1}^n l(y_i, f(\mathbf{x}_i)) + \text{pen}_\lambda(\mathbf{w}) \quad (13)$$

where the loss term is described by a sum of the hinge loss functions $l(y_i, f(\mathbf{x}_i)) = [1 - y_i f(\mathbf{x}_i)]_+ = \max(1 - y_i f(\mathbf{x}_i), 0)$ for each \mathbf{x}_i , $i = 1, \dots, n$. The penalty term is denoted as $\text{pen}_\lambda(\mathbf{w})$ and can have different forms, as mentioned before.

SVM with L_2 -penalization - Ridge penalty

The Ridge penalty uses the L_2 norm:

$$\text{pen}_\lambda(\mathbf{w}) = \lambda \|\mathbf{w}\|_2^2 = \lambda \sum_{j=1}^P w_j^2 \quad (14)$$

The L_2 penalty shrinks the coefficients to control their variance. However, the Ridge penalty provides no shrinkage of the coefficients to zero and hence no FS is performed.

SVM with L_1 -penalization - LASSO

The LASSO ('Least Absolute Shrinkage and Selection Operator'), which resorts to a L_1 penalization function, was originally proposed for generalized linear models by Tibshirani [48]. Later, Bradley and Mangasarian [44] adapted the L_1 -regularisation to SVM. Then, the penalty term has the form

$$pen_{\lambda}(\mathbf{w}) = \lambda \|\mathbf{w}\|_1 = \lambda \sum_{j=1}^p |w_j| \quad (15)$$

As a result of singularity of the L_1 penalty function, L_1 - SVM can automatically select features by shrinking the small coefficients of the hyperplane to zero. By using the L_1 norm penalization, the number of selected features is bounded by the number of samples and only one feature from a group of correlated features is selected, with the elimination of the others.

Smoothly Clipped Absolute Deviation SVM

The SCAD penalty is a non-convex penalty function firstly proposed by Fan [80] and then discussed by Fan and Li [81]. Later, Zhang et al. [7878] combined SVM with the SCAD penalty for FS. The penalization term for SCAD SVM has the form

$$pen_{\lambda}(\mathbf{w}) = \sum_{j=1}^p p_{SCAD(\lambda)}(w_j) \quad (16)$$

where the SCAD penalty function for each coefficient w_j is defined as

$$p_{SCAD(\lambda)}(w_j) = \begin{cases} \lambda |w_j| & \text{if } |w_j| \leq \lambda, \\ \frac{|w_j|^2 - 2a\lambda|w_j| + \lambda^2}{2(a-1)} & \text{if } \lambda < |w_j| \leq a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if } |w_j| > a\lambda, \end{cases} \quad (17)$$

with w_j , $j = 1, \dots, p$ as the coefficients defining the hyperplane and $a > 2$ and $\lambda > 0$ as tuning parameters. Fan and Li showed that SCAD prediction is not sensitive to selection of the tuning parameter a . Their suggested value $a = 3.7$ is therefore used as the default value. $p_{SCAD(\lambda)}(\mathbf{w})$ corresponds to a quadratic spline function with knots λ at and $a\lambda$. For small coefficients w_j , $j = 1, \dots, p$, the SCAD yields the same behaviour as the L_1 . However, for large coefficients, the SCAD applies a constant penalty, in contrast to the L_1 penalty, which increases linearly as the coefficient increases. The absolute maximum of the SCAD penalty, which is independent from

the input data, decreases the possible bias for estimating large coefficient. Furthermore, the SCAD penalty holds better theoretical properties than the L_1 penalty, as reported in [81].

Elastic Net

To overcome the limitations of LASSO, Zou and Hastie [77] proposed the Elastic Net, which consists in a linear combination of L_1 and L_2 penalties:

$$pen_{\lambda}(\mathbf{w}) := \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2, \lambda_1, \lambda_2 \geq 0 \quad (18)$$

The Elastic Net penalty provides automatic FS similar to L_1 , but is no longer bounded by the sample size. Moreover, at the same time this penalty manages the so called grouping effect, *i.e.* the selection of highly correlated features. Increasing λ_1 reduces the number of features of the classifier, whereas for large λ_2 one observes better control of the grouping effect. Wang [82] adapted the Elastic Net penalty to SVM classification problems. Therefore, the Elastic Net SVM optimization problem can be written as

$$\min_{b, \mathbf{w}} \frac{1}{n} \sum_{i=1}^n [1 - y_i f(x_i)]_+ + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2 \quad (19)$$

where $\lambda_1, \lambda_2 \geq 0$ are the corresponding tuning parameters.

Elastic Smoothly Clipped Absolute Deviation SVM

Fan and Li [81] showed the advantages of the SCAD penalty over the L_1 penalty. However, using the SCAD penalty might be too strict in selecting features for non-sparse data. A modification of the SCAD penalty analogously to Elastic Net could keep the advantages of the SCAD penalty, avoiding at the same time too restrictive sparsity limitations for non-sparse data. Therefore a combination of the SCAD and the L_2 penalties was proposed by Becker et al. [Errore. L'origine riferimento non è stata trovata.], which has the form

$$pen_{\lambda}(\mathbf{w}) := \sum_{j=1}^P p_{SCAD}(\lambda_1)(w_j) \lambda_2 \|\mathbf{w}\|_2^2 \quad (20)$$

where $\lambda_1, \lambda_2 \geq 0$ are the tuning parameters. The Elastic SCAD is expected to improve the SCAD method for less sparse data. According to the nature of the SCAD and L_2 penalties, the Elastic SCAD is expected to show good prediction accuracy for both sparse and non-sparse data.

The Elastic SCAD combined penalty provides sparsity, continuity, and asymptotic normality when the tuning parameter for the Ridge penalty converges to zero, *i.e.* $\lambda_2 \rightarrow 0$. The Elastic SCAD SVM optimization problem has the form

$$\min_{b, \mathbf{w}} \frac{1}{n} \sum_{i=1}^n [1 - y_i f(x_i)]_+ + \sum_{j=1}^p p_{SCAD(\lambda_1)}(w_j) + \lambda_2 \|\mathbf{w}\|_2^2 \quad (21)$$

where $\lambda_1, \lambda_2 \geq 0$ are the tuning parameters.

By solving Equation (21) the same problems as for SCAD SVM turn up, *i.e.* the hinge loss function is not differentiable at zero and the SCAD penalty is not convex in \mathbf{w} . The Elastic SCAD SVM objective function can be locally approximated by a quadratic function and the minimization problem can be solved iteratively similar to the SCAD approach [78; 81]. Renaming the SCAD penalty $p_{SCAD(\lambda_1)}(|w_j|)$ as $p_{\lambda_1}(|w_j|)$, the first-order derivative of the penalty is denoted by $p'_{\lambda_1}(\cdot)$. Denote the penalized objective function in Equation (21) by

$$A(b, \mathbf{w}) := \frac{1}{n} \sum_{i=1}^n [1 - y_i f(x_i)]_+ + \sum_{j=1}^p p_{\lambda_1}(|w_j|) + \lambda_2 \sum_{j=1}^p \|w_j\|_2^2$$

For each i (with respect to the fact that $y_i^2 = 1$) the loss term can be split according to

$$[1 - y_i(b + \mathbf{w} x_i)]_+ = \frac{1 - y_i(b + \mathbf{w} x_i)}{2} + \frac{|y_i - (b + \mathbf{w} x_i)|}{2}$$

Given an initial value (b_0, \mathbf{w}_0) close to the minimum of $A(b, \mathbf{w})$, the following local quadratic approximations is considered:

$$|y_i - (b + \mathbf{w} x_i)| \approx \frac{1}{2} \frac{\{y_i - (b + \mathbf{w} x_i)\}^2}{|y_i - (b_0 + \mathbf{w}_0 x_i)|} + \frac{1}{2} |y_i - (b_0 + \mathbf{w}_0 x_i)|$$

When w_{j0} is close to zero, set $\hat{w}_j = 0$; otherwise the approximation for the SCAD penalty is used

$$p_{\lambda}(|w_i|) \approx p_{\lambda}(|w_{j0}|) + \frac{1}{2} \frac{p'_{\lambda}(|w_{j0}|)}{(|w_{j0}|)} (w_j^2 - w_{j0}^2)$$

where $|w_j|$ is used instead of w_j due to the symmetrical nature of the SCAD penalty. Both approximations and their original functions have the same gradient at the point (b_0, \mathbf{w}_0) . Thus, the solution of the local quadratic function corresponds approximately to the solution of the original problem. Minimizing $A(b, \mathbf{w})$ is equivalent to minimizing the quadratic function

$$\tilde{A}(b, \mathbf{w}) = \frac{1}{2} \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix}^T Q \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix} - P \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix} \quad (22)$$

The solution to Equation (22) satisfies the linear equation system

$$Q \begin{pmatrix} \hat{b} \\ \hat{\mathbf{w}} \end{pmatrix} = P \quad (23)$$

Briefly, the Elastic SCAD SVM can be implemented by the following iterative algorithm:

1. set $k=1$ and specify the initial value $(b^{(1)}, \mathbf{w}^{(1)})$ by standard L_2 SVM according to Zhang et al. [78];
2. store the solution of the k^{th} iteration: $(b_0, \mathbf{w}_0) = (b^{(k)}, \mathbf{w}^{(k)})$;
3. minimize $\tilde{A}(b, \mathbf{w})$ by solving Equation (23) and denote the solution as $(b^{(k+1)}, \mathbf{w}^{(k+1)})$;
4. let $k = k + 1$. Go to step 2 until convergence. If elements $w_j^{(k)}$ are close to zero, e.g. smaller than 10^{-4} , then the j^{th} variable is considered to be redundant and in the next step will be removed from the model. The algorithm stops after convergence of $(b^{(k)}, \mathbf{w}^{(k)})$.

Tuning parameters are used to balance the trade-off between data fit and model complexity and they are usually determined by a grid search. The grid search method calculates a target value, e.g. the misclassification rate, at each point over a fixed grid of parameter values. This method may deal with local minima but it is not very efficient. The density of the grid plays a critical role in finding global optima. For very sparse grids, it is very likely to find local optimal points. By increasing the density of the grid, the computation cost increases rapidly with no guaranty of finding global optima. The major drawback of the fixed grid approach lies in the systematic check of the misclassification rates in each point of the grid, since there is no possibility to skip redundant points or to add new ones. When more parameters are included in the model, the computation complexity is increased and therefore the fixed grid search is only suitable for tuning of very few parameters. On the other hand, an interval search could be used, as suggested by Froehlich and Zell, who proposed an efficient algorithm of finding a global optimum on the tuning parameter space using a method called EPSGO (Efficient Parameter Selection via Global Optimisation) [83]. Using k -fold CV, the dataset is randomly split into k disjoint parts of roughly equal size (usually $k = 5$ or $k = 10$). In addition, the data is often proportionally split, *i.e.* each fold contains approximately the same distribution of class labels as the whole dataset (stratified CV). For each subset, the model is fitted using the other $k - 1$ parts and calculates the prediction error of the selected k^{th} part of the data. The choice of k determines a trade-off between bias and variance of the prediction error. Kohavi [84] showed that 10-fold stratified CV showed better performance in terms of bias and variance compared to $10 < k < n$. Hastie et al. [74] recommended to perform 5- or 10-fold CV as a good compromise between variance and bias. In the proposed pipeline, a 5-fold CV with interval search (default setting) for tuning parameters was preferred.

Generally, L_1 -SVM, SCAD SVM, Elastic Net SVM and Elastic SCAD SVM outperform ordinary L_2 -SVM using Ridge penalty. From the simulation study in [Errore. L'origine riferimento non è stata trovata.] emerged that FS methods with combined penalties are more robust to changes of the model complexity than using single penalties alone considering sufficiently large sample sizes. The SCAD SVM (followed by the L_1 -SVM) shows very good performance in terms of prediction

accuracy for very sparse models, but fails for less sparse models. Elastic Net and Elastic SCAD, which are based on combined penalty functions, show similar performance with respect to prediction accuracy and perform well for sparse and less sparse models. Both ‘elastic’ methods are able to consider correlation structures in the input data. However, the Elastic SCAD SVM in general provides more sparse classifiers than the Elastic Net SVM.

Moreover, SVM can be adapted to become a nonlinear classifier through the use of nonlinear kernels. While SVM is a binary classifier in its simplest form, it can function as a multiclass classifier by combining several binary SVM classifiers, *i.e.* creating a binary classifier for each possible pair of classes.

3.2 Second step: classifier development

In the second step of the proposed pipeline, several Linear SVM models [85] are implemented by varying SVM parameters and the number of included features on the basis of forward selection along the bootstrap generated list of the first step. To adjust for overoptimism, each model is validated with a LOOCV procedure [86] (see paragraph 3.2.2), using the Youden index [87] as measure of classification performance. The choice of the final model is based both on maximizing classification performance and minimizing the number of features included in the classifier (parsimony criterion).

The development of a classifier using high dimensional data is particularly challenging because firstly implies the choice between a classical (*e.g.* Penalized Logistic Regression model) or a machine learning algorithm (*e.g.* SVM models). In our pipeline, the well-known Linear SVM model was preferred to other more complex SVM models, since it is a simple model requiring the tuning of only two parameters:

- 1) the cost parameter, which controls the penalty imposed to the SVM model in case of misclassification of a training subject and therefore model complexity (the higher the cost, the less the misclassified subjects);
- 2) the class weights, indicating the influence assigned to the two classes (it should be used a fortiori in case of unbalanced groups of samples).

However, different kernel functions may be used for nonlinear SVM: the ‘custom’ ones (*i.e.* already implemented in R software) such as polynomial, Sigmoid, Gaussian Radial Basis Function (RBF) or user-defined kernels. The model choice and the selection of the ‘proper’ kernel function represent two issues to be further investigated. For the implementation of the Linear SVM model, the *svm* function included in the *e1071* package is used. The SVM output,

represented by the ‘scores’ (also indicated as ‘decisional values’), is transformed into probability estimates by means of the Platt’s scaling [88], a method invented by John Platt in the context of SVM, but that can be also applied to other classification models: a binary logistic regression is applied (fitted within a k-fold CV procedure), where the dependent variable is the class and the independent variable is the SVM score. This scaling produces two prediction errors, *i.e.* by the SVM model and by the logistic model. The CV involved in Platt’s scaling is an expensive operation for large datasets and the choice of k is another issue to be further explored, but conventionally the 5-fold CV is used. In addition, the probability estimates may be inconsistent with the scores, in the sense that the ‘argmax’ of the scores may not be the ‘argmax’ of the probabilities (*e.g.* in binary classification, a sample may be labeled by predict as belonging to a class that has probability < 0.5 according to predict probabilities). To conclude, the above probability estimates cannot be considered ‘true’ class probabilities (such as, for example, those directly estimated by using a binary logistic model), rather being ‘surrogated’ class probabilities. Therefore, it would be advisable not to transform the SVM scores into the class probabilities.

3.2.1 Linear Support Vector Machines

Linear SVM assumes that data are linearly separable. For a two-class classification problem on a dataset $S = \{x_i; y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^p$ is the feature vector of the i^{th} data point, and $y_i \in \{-1; 1\}$ is the corresponding label, the linear SVM classifier recovers an optimal separating hyperplane $\mathbf{w}^T \mathbf{x} + b = 0$ which maximizes the margin of the classifier. This can be formulated into the following constrained optimization problem [89; 90]:

$$\min_{\mathbf{w}, \gamma} \frac{\|\mathbf{w}\|^2}{2} + C \sum_i l(\mathbf{w}; x_i, y_i) \quad (24)$$

where $l\{\mathbf{w}; x_i, y_i\} = [1 - yf(x)]_+ = \max(1 - yf(x), 0)$ is the hinge loss, $f(x) = \mathbf{w}^T x_i - \gamma$ is the decision function, \mathbf{w} is the weight vector, γ is the bias term. SVM can also be trained by solving the Lagrangian dual of Equation (24), which results:

$$\begin{aligned} \max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i \alpha_i x_i^T x_j y_j \alpha_j \\ \text{s.t. } 0 \leq \alpha \leq C \quad \text{and} \quad \sum_i y_i \alpha_i = 0 \quad \forall i \end{aligned} \quad (25)$$

The classifier for linear SVM is then represented by

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad (26)$$

$$= \sum_{\alpha_i > 0} \alpha_i y_i \langle x_i, \mathbf{x} \rangle + b \quad (27)$$

The weight vector can be computed explicitly by

$$\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i x_i$$

and used for prediction. The advantage of solving the dual form is that only dot-products between data-points are needed. Consequently, the nonlinear SVM can then be trained by replacing the dot-products in Equation (25) with the corresponding kernel $K(x_i, x_j)$. The resulting classifier for the nonlinear SVM is then represented by

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{\alpha_i > 0} \alpha_i y_i K(x_i, \mathbf{x}) + b \quad (28)$$

where the α 's are the Lagrangian multipliers. Conceptually, the only difference between the nonlinear SVM and their linear counterparts is in the use of kernel function instead of the dot-product in Equation (27). Computationally, linear SVM can be directly evaluated by using Equation (26), which is much more efficient than nonlinear SVM for prediction purposes. Note that only those instances with positive value of α_i , *i.e.* the SV, will contribute to classification. As mentioned before (paragraph 3.1.3), different kernels can be used for nonlinear SVM.

The linear kernel is a special case of a degree- d polynomial kernel ($d=1$)

$$K(x_i, x_j) = (x_i^T x_j + c)^d$$

where $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial. When $c = 0$, the kernel is called homogeneous.

The Gaussian RBF kernel is the most widely used for nonlinear SVM

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

where $\gamma > 0$ is the scale parameter that controls the width of Gaussian, *i.e.* the decay rate of the distance. The parameter γ plays a similar role as the degree of the polynomial kernel in controlling the flexibility of the resulting classifier.

The Hyperbolic Tangent kernel, also known as the Sigmoid kernel, has the form

$$K(x_i, x_j) = \tanh(\alpha x_i^T x_j + c)$$

where there are two adjustable parameters, the slope α and the intercept constant c . A common value for α is $1/N$, where N is the data dimension.

Despite the effectiveness in classification of nonlinear data, the nonlinear SVM has much higher computational complexity for prediction than its linear counterpart. According to Equation (28), it requires $|V|d$ summations and $|V|$ exponential operations to compute the kernel function

values for evaluating the classifier output at a single testing instance, where d is the data dimension and $|V|$ is the cardinality of the set of SV. Prediction can be very time consuming, depending on the training data set size and the degree of nonlinearity. To cope with a highly nonlinear decision boundary, a great number of SV may be generated in the training process, which leads to higher computational complexity in testing time. Also, a larger training dataset will result in greater number of SV in training than a smaller dataset does. This also adds up to the complexity of prediction. Hence nonlinear SVM is inefficient for large-scale prediction tasks where there are thousands or millions of data points to classify in the validation set. Linear SVM, on the other hand, do not have such problem. It can perform prediction with d summations via Equation (26) and the testing time is independent of the number of SV. This makes it a much more efficient alternative than nonlinear SVM. Moreover, there are very efficient algorithms for training linear SVM which can run in linear time ([91]; [92]). This is impossible for nonlinear SVM, as the evaluation of the full kernel matrix is already quadratic in time complexity. Nevertheless, the performance of linear SVM is usually suboptimal as compared to nonlinear ones since they cannot handle linearly inseparable data. It would be desirable to develop an SVM model with both the efficiency of the linear SVM at the prediction stage and the classification accuracy of the nonlinear SVM classifier [93].

3.2.2 Cross Validation procedure

In data mining, a typical task is to learn a model from available data. The problem with evaluating such a model is that the classification rule developed from the dataset of known data on which training is run (training dataset) might fail in predicting unseen data (validation dataset). CV, sometimes called rotation estimation ([94]; [95]; [96]), is a procedure for estimating the generalization performance in this context. The idea for CV originated in the 1930s [97]. Mosteller et al. [98] developed the idea. A clear statement of CV, which is similar to current version of k -fold CV, first appeared in [99]. In 1970s, both Stone [86] and Geisser [100] employed CV as means for choosing proper model parameters, as opposed to using CV purely for estimating model performance. Currently, CV is widely accepted in data mining and machine learning community, and serves as a standard procedure for performance estimation and model selection. There are two possible goals in CV:

- to estimate performance of the learned model from available data using one algorithm, *i.e.* to gauge the generalizability of an algorithm;
- to compare the performance of two or more different algorithms and find out the best algorithm for the available data, or alternatively to compare the performance of two or more variants of a parameterized model.

The above two goals are highly related, since the second goal is automatically achieved if one knows the accurate estimates of performance. One of the main reasons for using CV instead of using the conventional validation (e.g. partitioning the dataset into two sets of 70% for training and 30% for validation) is that the error (e.g. Root Mean Square Error) on the training set in the conventional validation is not a useful estimator of model performance and thus the error on the validation data set does not properly represent the assessment of model performance. This may be because there is not enough data available or there is not a good distribution and spread of data to partition it into separate training and validation sets. In these cases, a fair way to properly estimate model prediction performance is to use CV as a powerful general technique [101]. Indeed, CV combines (averages) measures of fit (prediction error) to correct for the optimistic nature of training error and derive a more accurate estimate of model prediction performance.

Various procedures of CV have been proposed, such as resubstitution validation, Hold-Out Validation (HOV), k-fold CV, LOOCV and repeated k-fold CV.

Resubstitution validation

In resubstitution validation, the model is learned from all the available data and then tested on the same set of data. This validation process uses all the available data but suffers seriously from data overfitting. Therefore, the algorithm might perform well on the available data yet poorly on future unseen data.

Hold-Out Validation

To avoid data overfitting, an independent validation set is preferred. A natural approach is to split the available data into two parts: one for training and the other for validation. The validation data is held out and not looked at during training. HOV avoids the overlap between training data and validation data, yielding a more accurate estimate for the generalization performance of the algorithm. One drawback is that this procedure does not use all the available data and the results are highly dependent on the choice for the training/validation split. The instances chosen for inclusion in the validation set may be too easy or too difficult to classify and this can skew the results. This issue can be partially addressed by repeating HOV multiple times and averaging the results, but unless this repetition is performed in a systematic manner, some data may be included in the validation set multiple times while others are not included at all, or conversely some data may always fall in the validation set and never get a chance to contribute to the learning phase. Therefore, k-fold CV is used to deal with these challenges and fully utilize the available data.

K-fold Cross Validation

In k-fold CV the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is hold-out for validation while the remaining k - 1 folds are used for learning. Data is commonly stratified prior to being split into k folds. Stratification is the process of re-arranging the data as to ensure that each fold is a good representative of the whole.

Leave-One-Out Cross Validation

LOOCV is a special case of k-fold CV where k equals the number of instances in the data (*i.e.* $k = n$). In other words in each iteration nearly all the data except for a single observation are used for training and the model is tested on that single observation. An accuracy estimate obtained using LOOCV is known to be almost unbiased but it has high variance, leading to unreliable estimates [102102]. It is still widely used when the available data are very rare, especially in bioinformatics where only dozens of data samples are available.

Repeated k-Fold Cross-Validation

Large number of estimates are always preferred in order to obtain reliable performance estimation. In k-fold CV, only k estimates are obtained. A commonly used method to increase the number of estimates is to run k-fold CV multiple times. The data is re-shuffled and re-stratified before each round.

In the proposed pipeline, which was applied in the context of bioinformatics, the LOOCV was used, since very often these types of data are characterized by a small amount of samples.

3.3 Presentation of results

A good presentation of results should be immediate and easy to understand. To this aim, innovative plots are proposed to graphically summarize the results obtained in each step of the procedure used for class prediction.

3.3.1 The ‘egg-shaped’ plot

An example of the egg-shaped plot representation of top ranking features is shown in **Figure 9**. The nodes and edges represent, respectively, feature bootstrap occurrences and co-occurrences (pairwise occurrences). The larger the node, the more often the corresponding feature occurs; the thicker the edge between two features, the more often they are selected together in bootstrap samples. The plot can be filtered to show features with co-occurrences in the

bootstrap samples at least equal to a certain threshold. We report here an example of ‘egg-shaped’ plot, where the feature miR-451 emerges as the most important feature, followed by miR-16, miR-486-5p, miR-92a and miR-22. The most selected features are also the most interconnected: the strongest co-occurrence involves miR-451 connected to miR-16, followed by miR-486-5p and miR-92a. Also miR-16 presents several interconnections with miR-92a, miR-486-5p, and miR-22.

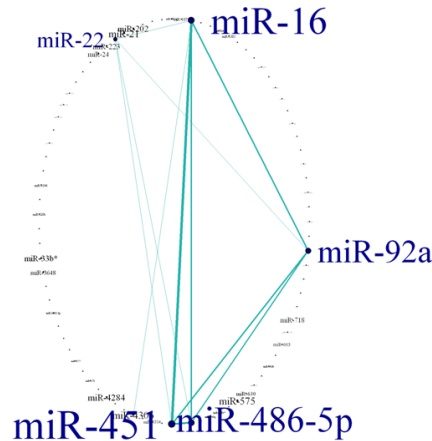


Figure 9. Example of egg-shaped plot of the class prediction results (first step).

3.3.2 The ‘ROC space’ plot

The LOOCV SVM performances can be graphically summarized in terms of False Positive Rate (FPR) and True Positive Rate (TPR) by implementing a ‘ROC space’ plot, which can be seen as a generalization of the ROC curve (an example is shown in Figure 10). The best performing groups of models are those nearest to the point (FPR=0, TPR=1) of the ‘ROC space’.

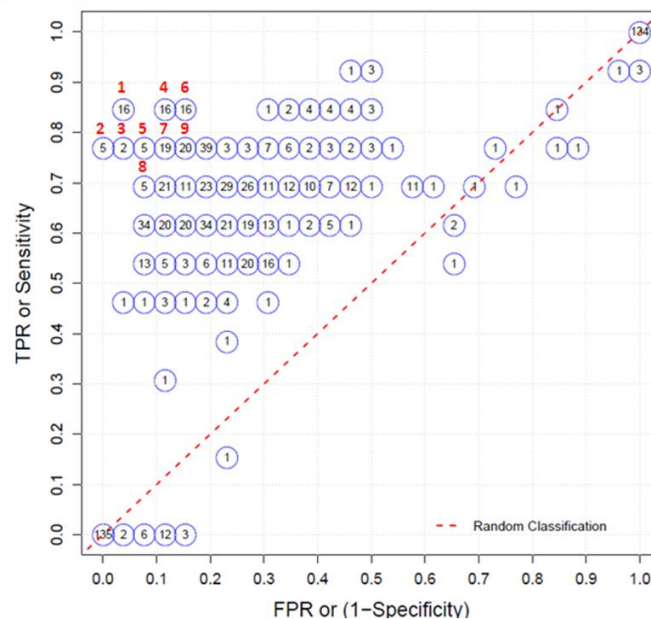


Figure 10. Example of ROC space plot of the class prediction results (second step). The best performing groups of linear SVM models are indicated by an ID number outside the circles; inside each circle is reported the number of models produced by the LOOCV procedure corresponding to a specific classification performance.

3.4 The Cineca system

Cineca is a non-profit consortium, made up of 70 Italian universities, four national research centres, and the Ministry of Universities and Research (MIUR), established in 1969 in Casalecchio di Reno, Bologna. In 2013 it was the most powerful supercomputing centre for scientific research in Italy, ranked at the 23rd position in 2014 and at the 32nd position in 2015 [103]. The institutional mission of the consortium is to support the Italian scientific community through supercomputing and scientific visualization tools. Since the end of the 1980s, Cineca has broadened the scope of its mission by embracing other IT sectors, developing management and administrative services for universities and designing ICT systems for the exchange of information between the Ministry of Education, MIUR and the Italian national academic system. The consortium is also strongly committed to transfer technology to many categories of users, from public administration to the private enterprises. Moreover, Cineca takes part in several research projects funded by the European Union for the promotion and development of IT technologies (grid computing, bioinformatics, digital content, the promotion of transnational access to European supercomputing centres, etc.). Cineca can be seen as a high technology bridge between the academic world, research and the world of industry and public administration.

The Cineca system was used to take advantage of its powerful hardware in order to perform computationally intensive analyses, involved in the implementation of the Elastic Smoothly Clipped Absolute Deviation SVM (Elastic SCAD SVM) or, alternatively, the SVM with L_1 -penalization (L_1 -SVM) algorithms.

4. A comprehensive pipeline for class comparison and class prediction

Taking into account the results of the simulation study on two-sample tests for class comparison and the methodological insights on the classification algorithms for class prediction, a comprehensive pipeline for class comparison and class prediction (Figure 11) was developed with innovative applications of existing bioinformatics methods including:

- 1) the combination of the results of two statistical tests (t and AD) to detect features with significant fold change or general distributional differences in class comparison;
- 2) the application of a bootstrap selection approach together with machine learning techniques included in a Leave-One-Out Cross Validation (LOOCV) procedure to guarantee result generalizability and study the interconnections among the selected features in class prediction.

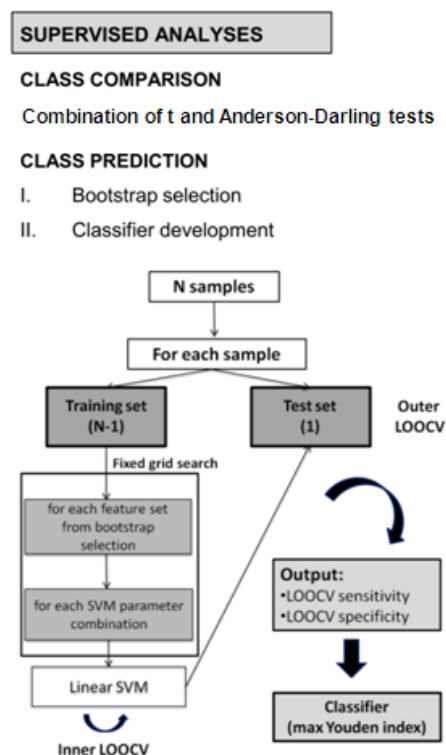


Figure 11. Workflow for class comparison and class prediction analyses.

The presented pipeline may be extended to other kinds of 'omics' studies by introducing proper methodological adjustments. For instance, with small sample sizes, the Wilcoxon-Mann-Whitney test should be used for class comparison analysis; for class prediction analysis, with non-coding

RNA Sequencing data, which are count variables, models suitable for analyzing count data should be used (*i.e.* Negative Binomial, Poisson distribution based models).

5. Real data applications

We applied the proposed pipeline to two different high dimensional datasets, *i.e.* plasmatic miRNA data [10] and SESI-MS data [104].

5.1 Plasmatic microRNA data

Proposal of supervised data analysis strategy of plasma miRNAs from hybridisation array data with an application to assess hemolysis-related deregulation

Abstract

Background: Plasma miRNAs have the potential as cancer biomarkers but no consolidated guidelines for data mining in this field are available. The purpose of the study was to apply a supervised data analysis strategy in a context where prior knowledge is available, *i.e.* that of hemolysis-related miRNAs deregulation, so as to compare our results with existing evidence.

Results: We developed a structured strategy with innovative applications of existing bioinformatics methods for supervised analyses including: 1) the combination of two statistical (t- and Anderson-Darling) test results to detect miRNAs with significant fold change or general distributional differences in class comparison, which could reveal hidden differential biological processes worth to be considered for building predictive tools; 2) a bootstrap selection procedure together with machine learning techniques in class prediction to guarantee the transferability of results and explore the interconnections among the selected miRNAs, which is important for highlighting their inherent biological dependences. The strategy was applied to develop a classifier for discriminating between hemolyzed and not hemolyzed plasma samples, defined according to a recently published hemolysis score. We identified five miRNAs with increased expression in hemolyzed plasma samples (miR-486-5p, miR-92a, miR-451, miR-16, miR-22).

Conclusions: We identified four miRNAs previously reported in the literature as hemolysis related together with a new one (miR-22), which needs further investigations. Our findings

confirm the validity of the proposed strategy and, in parallel, the hemolysis score capability to be used as pre-analytic hemolysis detector.

Background

MicroRNAs (miRNAs) are highly conserved single-stranded small RNA molecules (~19-22 nucleotides long) that play a key role in post-transcriptional gene regulation. To date, more than 2600 human miRNAs have been identified (miRBase V21 [1]). This class of small RNAs is being widely studied in oncology and a functional implication in cancer development and progression has been demonstrated [2,3,4]. Recent studies have shown that miRNAs can be released from cells (encapsulated in exosomes and/or bound to proteins and lipoproteins) and enter into the circulation as a consequence of an active release or apoptotic and necrotic cell death [5,6,7,8]. As a result of miRNA release from cells, these molecules have also been found in every human body fluid, in a stable form protected from endogenous RNases, thus making plasma miRNA levels well suited for non invasive analysis in patient samples [9,10]. Independent studies have reported the feasibility of using plasma miRNAs as promising disease biomarkers and, in the context of malignancies, they have shown a potential as molecular tools for detection, prognosis and treatment decision making of various cancers [11,12]. However, some biological or technical challenges could limit the development of this class of biomarkers [13,14], thus probably giving an explanation of the poor concordance among inter-study results [15].

In the attempt to develop a multimarker classifier using plasma miRNA data, some issues arising during the discovery process challenge the researchers. Moreover, so far there are no consolidated guidelines for data analysis in this context. This prompted us to develop a structured strategy for supervised analyses with the aim of: (1) in class comparison analysis, detecting differences of miRNA distributions between the two compared classes; (2) in class prediction analysis, discovering the top discriminating features, study their associations and interconnections, and developing a 'robust' cross validated classifier. In the class comparison step we proposed the combined use of two tests: the t-test and the nonparametric Anderson-Darling (AD) test [16]. The former is commonly applied for class comparison being directly related to the fold change (FC), which is taken as a measure of the 'differential expression' direction and strength; however, the FC is limited to the exploration of differences between the mean expression values in the two compared classes. On the other hand, the AD test is able to detect more general differences between two classes, which could reveal hidden differential biological processes. In class prediction we set up an assumption-free procedure for the development of a cross validated classifier, after a robust miRNA ordering via bootstrap sampling.

The above approaches were applied to plasma miRNAs determined on a subset of patient samples from a clinical trial series [17]. RNA extracted from these samples was subjected to

Agilent miRNA hybridization array. A microarray approach was chosen because it allows reaching a higher throughput than PCR-based assays (even if it is able to analyze only miRNAs already known and annotated in miRBase [18]) and is expected to be advantageous in a discovery phase. Different miRNA microarray platforms, able to measure circulating miRNAs, are commercially available, including GeneChip miRNA Array by Affimetrix, Human miRNA Microarray by Agilent. Among these, we opted for the Agilent system, since it emerged as one of those obtaining the highest performances and is probably the most commonly used. In addition, in a pilot study that we have recently published, the feasibility of using such a platform in miRNA detection also from archival plasma samples was evaluated [19] and we found a very high correlation between technical replicates and a good correlation between different batches. We focused on the comparison between miRNA expression profiles from hemolyzed and not-hemolyzed plasma samples, thus choosing a context where prior knowledge on deregulated miRNAs is available.

Methods

The strategy that we developed for data preparation and data analysis is illustrated in Additional Figure 1. All analyses were performed using R and in particular Bioconductor libraries [20]. The details are reported below.

Study design

Plasma samples included in the present study come from patients entering a randomized breast cancer prevention Trial [17]. In details, we analyzed a subset of patients from the group of 1476 patients enrolled in the control (not treated) arm of the trial at the Fondazione IRCCS Istituto Nazionale dei Tumori. Blood samples, collected using heparin, were separated into plasma aliquots by centrifugation (2000xg; 15 min at 4°C) and stored at -80°C until assayed; no thawing accident occurred during storage. Since the blood samples were collected for different purposes, no information are available on erythrocyte or platelet counts. Nevertheless, the presence of hemolysis was evaluated in the plasma samples on the basis of the 'Hemolysis Score' (HS) previously published by our group [21]. Our 'controls' were not-hemolyzed plasma samples ($HS \leq 0.057$) and our 'cases' were the samples with $HS > 0.14$, roughly corresponding to a visible hemolysis. The remaining samples showing $0.057 < HS \leq 0.14$ were not analyzed. As cases and controls could be unbalanced for some variables, a matching procedure was used, by applying the nearest neighbor matching within specified propensity score (PS) calipers [22] in order to have a more relaxed criterion which would enable us to match all the hemolyzed samples. Given the PS, that is the probability of assignment to one group conditional on some characteristics of patients and samples (*i.e.* disease status, age at drawing and drawing year), we matched each case with two controls with the closest PS within the specified range (the caliper width). We used the recommended caliper width, which is equal to the 20% of the standard deviation of the

PS logit [23]. After matching we randomly split the sample in half into a training set for supervised analyses and a validation set for internal validation of results, maintaining the 1:2 ratio between cases and controls.

Sample processing

Plasma isolation and RNA extraction were carried out as previously described [19]. Briefly, total RNA was extracted from 350 μ l plasma collected in heparin using the commercial column-based system Qiagen miRNeasy R Mini Kit (Qiagen, Valencia, CA, USA), slightly modified. Briefly, 400 μ l of plasma/medium were thawed on ice and centrifuged at 1000 \times g for 5 min in a 4 °C microcentrifuge. An aliquot of 350 μ l of plasma per sample was transferred into a new microcentrifuge tube and 1300 μ l of a Qiazol mixture containing 1.25 μ g/ml of MS2 bacteriophage RNA (Roche Applied Science, Milan, Italy) and a RNA spike-in (ath-miR-159a) to be able to eventually test the recovery efficiency by RT-PCR analysis. A rinse step (500 μ l Qiagen RPE buffer) was repeated 3 times. Total RNA was eluted by adding 25 μ l of RNase-free water to the membrane of the spin column and incubating for 1 min before centrifugation at 15,000 \times g for 1 min at room temperature. The heparin contained in the RNA samples was digested using heparinase I (Sigma- Aldrich, St. Louis, MO, USA), in the presence of an RNase inhibitor, (RNAsin; Promega, Madison, WI, USA) for 1 hour at room temperature, and RNA was stored at -80°C. The heparinase digestion was performed to make RNA suitable for downstream RT-PCR analysis (not pertinent to this paper, manuscript in preparation). In fact, For many years, the use of heparin for blood collection has been avoided in case of subsequent RNA extraction, since the anticoagulant inhibits PCR amplification [24, 25, 26, 27]. However, we have recently demonstrated that if adequately treated with heparinase, plasma samples derived from blood collected with heparin tubes are suitable for miRNA expression analysis, without affecting miRNA detection [28]. Hybridization on Agilent Human miRNA microarrays was carried out by Functional Genomics facility according to the manufacturer's instructions as previously described [19]. Briefly, SurePrint G3 Human v16 miRNA 8x60K microarrays (G4870A) designed on miRBase 16.0 from Agilent Technologies were used. 2.5 μ l of total RNA was dephosphorylated at 37°C for 30 min with calf intestinal phosphatase and denatured using 100% DMSO at 100°C for 5 min. Samples were labeled with pCp-Cy3 using T4 ligase by incubation at 16°C for 1 hour and hybridized. Arrays were washed according to manufacturer's instructions and scanned at a resolution of 5 μ m using an Agilent 4000B scanner. Data were acquired using Agilent Feature Extraction software version 10.7.

Data pre-processing

Raw data were summarized as previously described [19]. Briefly, in the employed platform, each miRNA is targeted by one to four different probes and each probes spotted 10 to 40 times on the array. Then, the total signal for each miRNA was obtained by summing the probe signals

derived from Agilent Feature Extraction software. Using this software, each probe is defined detected if its value is greater than three times its standard error, and each miRNA is defined as detected if at least one of the probes is detected. Summarized data were \log_2 transformed. Only the 1205 human ('hsa') miRNAs were considered in subsequent analyses. Microarray data are MIAME compliant and were deposited into the NCBI's Gene Expression Omnibus (GEO) database with accession number 'GSE59993' [29]. MiRNAs were filtered at 90%, *i.e.* we retained only miRNAs detected in at least 90% of all samples. By applying a less stringent filtering (*i.e.* 10% filtering), no additional differentially expressed (DE) miRNAs could be identified (data not shown), as compared with those obtained with the 90% filtering.

As regard to the normalization step, we applied the ratio-based approach [30] that is like using, in turn, all miRNAs as normalizers but eliminating any duplications, *i.e.* each miRNA pair only appeared once.

Supervised data analyses

We implemented supervised approaches for class comparison and class prediction on the training set samples using both raw (not normalized) and ratio-normalized data. Class comparison analysis, aimed at identifying features (miRNAs or miRNA ratios) DE between cases and controls, was based on the combined use of the t- and the non parametric AD [16] tests. While the t-test considers only location differences, the AD test is an 'omnibus test' [31], *i.e.* it considers the whole feature distribution, granting more importance to the observations in the tails. The latter characteristic becomes valuable when one is interested in finding signals that are only present in patient subsets diverging from the center of the distribution.

Moreover, plasma miRNA data, like other 'omics' data, have often not normal distributions and the sample sizes are often small. In presence of distributions with asymmetries, multimodality or heavy tails, the AD test reveals useful for the identification of interesting features. We considered the asymptotic version of the AD test, with correction for the presence of ties. The Benjamini-Hochberg method was used to distinctly adjust t- and AD p-values in order to control for the False Discovery Rate (FDR) [32]. In particular, we combined the results of the two tests by considering as significantly DE the features for which the FDR-adjusted p-value was below the 5% level for at least one of the two tests. This procedure could inflate the overall Type I error; however, we expect such an effect to be marginal because the two tests statistics are likely to be dependent and, in addition, both tests are applied to the same data.

For class prediction analysis, aimed at developing a classifier able to accurately discriminate between hemolyzed and not-hemolyzed samples, a two-step procedure was set up: firstly, with the purpose of obtaining a robust ranking of features with distributional differences between the two classes, a 'bootstrap selection' was performed, according to the strategy proposed by Austin and Tu [33]. We extracted 1000 bootstrap samples [34] and we applied three machine

learning selection algorithms, *i.e.* Prediction Analysis for Microarrays (PAM) [35], Random Forests (RF) with Boruta feature selection method [36] and Elastic Smoothly Clipped Absolute Deviation (SCAD) Support Vector Machines (SVM) [37], while maintaining the same proportion of hemolyzed and not-hemolyzed in each group. The three methods were chosen because they overcome the 'curse of dimensionality' usually present in high-dimensional data (*i.e.* more features than subjects) and because they are conceptually different algorithms that we considered as 'representative' of methodological categories using different decision rules for classification (*i.e.* a nearest centroid, a decision tree and a SVM based method, respectively). PAM, being characterized by a minor complexity respect to the other two algorithms, may be insufficient to appreciate complex classification patterns. Among the other two more sophisticated methods, RF overcome the main disadvantage of decision trees methods, which is their tendency to data overfitting and, like PAM, are fast and nonparametric, so one has not to worry about outliers. On the other hand, RF only output measures of feature importance, the interpretation of which is controversial with correlated features [38]. The inherent biological dependence among the features, which implies correlation among miRNAs, was taken into account by using the Elastic SCAD SVM algorithm. The features were ranked on the basis of the frequency of simultaneous selection by the three above algorithms, discarding the features not selected in at least one bootstrap sample. None of the three algorithms is uniformly superior in detecting class differences. Our strategy seeks to overcome the above limitation by implicitly relying on an intersection criterion, by which a feature emerges as 'strong' regardless of the statistical technique used for analysis. As second step, aimed at developing a cross validated classifier, we implemented a linear SVM model [39], using the features previously ranked according to the bootstrap selection. We chose the linear SVM since it is a simple model requiring only the tuning of two parameters, *i.e.* the cost, which controls model complexity and the class weights, indicating the influence assigned to the two classes. Different models were fitted by varying the number of included features, forwardly selected according to the bootstrap generated list. The models were then cross validated with a leave-one-out cross validation procedure [40] to adjust for overoptimism the classification performance measures, *i.e.* sensitivity, specificity and Youden index [41]. The final model used for developing the classifier was chosen according to both the criteria: best classification performance, measured by the highest Youden index, and smallest number of features included in the model. Finally, the classification performance measures of the chosen models were calculated on the validation set, together with their corresponding bootstrap 95% confidence intervals (CI) taken as an estimate of the performance measure variability.

Results and discussion

Sample processing and data pre-processing

After case-control matching, 78 samples were selected, 26 hemolyzed and 52 not-hemolyzed; 39 samples (13 hemolyzed vs 26 not-hemolyzed) were included in the training and validation set, respectively. After the filtering performed on the training set, 88 miRNAs were retained, based on which a total of 3828 ratios were generated.

Class comparison

The results of class comparison using raw and ratio data with the lists of miRNAs significantly DE according to t- or AD test, after adjusting for multiple testing were graphically summarized via volcano and concordance plots (Additional Figure 2). Concerning raw data, four miRNAs (4.5%) were significant at the t- or AD test. Three miRNAs (miR-486-5p, miR-92a, miR-451) were identified as up-regulated in hemolyzed samples through the t-test (Additional Figure 2A), being also detected by the AD test, as shown in the second quadrant of the concordance plot in the Additional Figure 2B (the adjusted p-values were coincident). Moreover, one more miRNA (miR-16) was significant according to the AD test alone (Additional Figure 2B), although the t-test p-value was near to the significance threshold. Regarding ratio data, 224 miRNA ratios (5.8%) were significant at the t- or AD test. We detected 104 ratios as significantly up-regulated and 94 ratios as significantly down-regulated with the t-test, for a total of 198 ratios, which involved 80 miRNAs (Additional Figure 2C). One hundred and seventy ratios (involving 68 miRNAs, including the four previously selected with raw data) were detected by both tests (first quadrant of Additional Figure 2D), 28 ratios (involving 27 miRNAs) only by the t-test (second quadrant of Additional Figure 2D) and 26 ratios (involving 29 miRNAs) only by the AD test (fourth quadrant of Additional Figure 2D). The features significantly DE in the training set at the raw and ratio data analysis were also evaluated in the validation set. All the 4 miRNAs and 203 over 224 ratios resulted DE in the validation set for the t- or the AD test (Additional Table 2).

Class prediction

Figure 1 summarizes the results of the first step of class prediction analysis with raw data ('bootstrap selection'). In particular, in Figure 1A the miRNAs are ranked according to the number N of occurrences in the bootstrap samples, *i.e.* the number of times in which they are jointly selected by the three machine learning algorithms. miRNAs identified in class comparison analysis as significantly up-regulated in hemolyzed samples resulted at the top positions of bootstrap ranking (top 35 miRNAs in Figure 1A). MiR-451 headed clearly in class prediction, being selected in 846 out of 1000 bootstrap samples, followed by miR-16 (779/1000), miR-486-5p (734/1000), miR-92a (668/1000) and miR-22 (448/1000). An egg-

shaped plot representation of top ranking miRNAs is shown in Figure 1B, where node size and edge thickness are proportional to the frequency of miRNAs occurrences and co-occurrences (pairwise occurrences) in the bootstrap samples; a filtering was applied to show only those miRNAs with co-occurrences at least equal to 300. The most frequent co-occurrences are shown in Figure 1C. Generally, the most selected miRNAs were also the most interconnected. In fact, considering miR-451, the strongest co-occurrence involved miR-16, being the two miRNAs jointly selected in 711 out of 1000 bootstrap samples, followed by miR-486-5p (624 co-occurrences) and miR-92a (606 co-occurrences). Also miR-16 presented several interconnections with miR-92a (604 co-occurrences), miR-486-5p (587 co-occurrences), and miR-22 (411 co-occurrences). MiR-451, miR-16, miR-486-5p and miR-92a have been previously reported in the literature as hemolysis-related plasma miRNAs [20], while miR-22 was selected in a high number of bootstrap samples and linked to the top four miRNAs. Ratio data generally led to smaller bootstrap occurrences, since each miRNA appeared in several ratios. However, miR-486-5p, miR-92a, miR-451 and miR-16 were included in the top eight ratios, with occurrences equal to 357 (1st position), 304 (2nd position), 270 (4th position) and 214 (8th position), respectively. MiR-22 appeared at the 31st position, with 121 occurrences. The 'autoselected' specific normalizers were miR-4257 for miR-486-5p and miR-92a, and miR-4286 for miR-451 and miR-16. The top co-occurrence involved miR-92a/miR-4257 and miR-486-5p/miR-4257, with a frequency equal to 200.

As regard to the classifier development (step 2), the 'ROC space' plot in Figure 2 summarizes the SVM model performance in terms of false positive rate (FPR) and true positive rate (TPR); as true for the ROC curves, ideal models are those closest to the point (0,1), corresponding to 100% sensitivity and specificity. The numbers inside the circles count the models with a specific combination of FPR and TPR, while the numbers outside (ID) rank each group of models in terms of performance, as quantified by the Youden index (e.g., ID=1 indicates the group of models with the highest Youden index). Considering raw data (Figure 2A), we identified 8 best performing groups; among them, 16 models (ID=1) showed the highest Youden index equal to 0.81. Using ratio data (Figure 2B) only one model stood alone in leading the rank classification list, with a Youden index of 0.73.

The above results are numerically shown in Table 1 (left panel) only for the best performing groups, *i.e.* those ID numbered in the Figure 2; additionally, for the specific model chosen in each group according to a parsimony criterion (smallest number of features), we show the parameters (middle panel) and the performance evaluated in the validation set (right panel). Considering raw data, the Youden index ranged from 0.61 to 0.81 in the training set and from 0.46 to 0.73 in the validation set. Among the 16 models with ID=1 (Youden index= 0.81), the chosen one included 35 miRNAs (Figure 1A). However, an alternative choice could be the one selected within the ID=8 group (Youden index=0.61), which included three miRNAs, *i.e.* miR-

451, miR-16 and miR-486-5p; such a model achieved the highest classification performance in the validation set (Youden index= 0.73).

Regarding ratio data, the Youden index ranged from 0.61 to 0.73 in the training set and from 0.58 to 0.77 in the validation set. The chosen model included 500 ratios (Youden index=0.73), corresponding to 88 features. Alternative choices could be the model with ID=8, including two ratios (miR-486-5p/miR-4257 and miR-92a/miR-4257) or that with ID=5, including 4 ratios (miR-486-5p/miR-4257, miR-92a/miR-4257, miR-486-5p/miR-4286, miR-4286/miR-451), the latter presenting a slightly better classification performance in the training set (Youden index=0.65 vs 0.61); also in this case, the two parsimonious models had the best performance in the validation set (Youden index=0.77). It is worth to notice that with ratio data the miR-16 would not have been selected, since the top ratios contained more than once the other hemolysis-related miRNAs, producing redundancy in the results.

Globally, we noticed that the SVM cost parameters, which control model complexity, were smaller with the ratio data and that, regardless the type of data, it was more difficult to validate a model containing a large number of miRNAs. Moreover, in the validation set the Youden index showed wide bootstrap confidence intervals (CI), due to the small sample size.

Conclusions

In the present work we developed a general analysis strategy in order to deal with some issues arising in the supervised analyses of plasma miRNA from hybridization array data. In the data pre-processing step, any normalization method can be applied and does not preclude the subsequent conduction of supervised analyses, although contributing to the final results. The normalization method should be chosen in relation to the type of features, their precision level and to the domain knowledge (*e.g.* possible availability of housekeeping features). While in our investigation we adopted a joint analysis of raw and ratio-normalized data, other methods might be suitable, like for instance the quantile method, previously shown to work best in reducing differences in miRNA expression values for tissue samples [42]. We just considered inappropriate the application of the global mean method, which would artificially produce down-regulated miRNAs. Such a problem was clearly demonstrated in the case of an expected general miRNA down-regulation as a consequence of inducible deletion of Dicer1 [43]. This is in contrast with the expectation of a global miRNA up-regulation in patients with cancer as a consequence of a passive (*i.e.* cancer cell death) or active (*i.e.* by microvesicles) release in bloodstream. To establish which miRNA in a ratio has relevant discriminating role and which act as normalizer (no modulation, *i.e.* FC=1, or presenting weaker modulation) the results of raw and ratio data analyses should be interpreted together. An advantage of the ratio method is that, in the absence of known housekeeping miRNAs, it allows identification and automatic handling of a specific normalizer for each DE miRNA.

In class comparison analysis, the search for DE features is usually intended for detecting significantly different means in the two groups, and location tests, such as the t-test, are commonly applied; this classifies class comparison analysis in the domain of univariable statistical analyses. However, the t-test assumption of normality is often not fulfilled when dealing with plasma miRNA data, mainly due to the skewed, heavy-tailed or multimodal distributions of expression values, especially if associated with small sample size. Moreover, focusing only on location, the t-test could miss miRNAs with a signal translating into more general differences between the distributions. Our strategy of combining the results of t- and AD tests was aimed at taking advantage of their different characteristics and allowed us to discover those miRNAs discarded by the t-test due to not significant FC, but with not overlapping feature distributions. The AD test is particularly valuable when distributions differ in the tails, which could reveal underlying biological differences. Class comparison analysis is a useful tool for detecting DE features; however, in our opinion caution should be taken in using it for ranking purposes. Indeed, by using the bootstrap selection in the first step of class prediction analysis, together with the application of the three machine learning algorithms (Elastic SCAD SVM, RF, PAM), more robust and possibly generalizable results can be obtained. Together with the bootstrap selection, we want to point out the egg-shaped plot, which can be used as a tool for giving an insight of interconnections among the selected features, becoming useful for highlighting their inherent biological dependences.

In the second step of the class prediction analysis, the classifiers are obtained by using statistical models including subgroups of selected features, and this categorizes class prediction in the domain of multivariable statistical analyses. The joint use of bootstrap selection and classifier cross validation should ensure the robustness of the class prediction results. A limitation of the procedure is that we could identify several best models in terms of classification performance. In some cases (especially using ratio data) the best models included a large number of features, thus being more prone to overfitting. However, we observed that the use of a small number of strongly predictive features resulted in a non significant decay of the cross validated classification performance measures in the testing set. Therefore, our strategy was to choose more parsimonious models, since it is likely that the features included in such models will not be filtered out during the data pre-processing step. However, our results have to be taken with caution due to the small sample size, as it emerged from the large bootstrap intervals of the classification performance measures. By using our strategy we identified four top miRNAs (miR-486-5p, miR-92a, miR-451, miR-16) that have been reported in the literature as related to the presence of hemolysis, together with another one (miR-22), which is worth to further investigate. Even though miR-22 was not directly described as hemolysis-susceptible miRNA, it was identified as a signature miRNA for erythrocyte maturation [44]. In addition, very recently MacLellan et al., by mimicking hemolysis through mechanical

lysis of blood samples in healthy individuals, found higher levels of serum miR-22 in lysed compared to matched unlysed samples ([45], Figure 1). Regarding the top miRNAs, we obtained consistent results in class comparison and bootstrap selection; indeed, strong signals are detectable on both raw and ratio data, even with univariable and not cross validated analyses. However, univariable methods unavoidably discard features that would have provided useful information, if taken in aggregate. More subtle differences, like those we observed for miR-22, could justify the use of more sophisticated methods, such as the bootstrap selection joined with the machine learning algorithms. The concordance of our results with literature data also corroborated the ability of the HS to discriminate between hemolyzed and not-hemolyzed samples and thus its usefulness as a pre-analytic hemolysis detector.

Classifier development should rely on availability of three distinct datasets for training, validation, and testing. We are aware that a limitation of the present study is the lack of availability of a testing set on which an unbiased assessment of classifier performance could be obtained. Unfortunately, three-fold splitting was not applicable in our case study, because was hampered by the small number of hemolyzed samples, and suitable public datasets (*i.e.*, data from Agilent miRNA hybridization array coupled with hemolysis score evaluation) were still unavailable.

Our strategy may be extended to other kinds of 'omics' studies by introducing proper methodological adjustments. For instance, with non-coding RNA Sequencing data, which are count variables, the Anderson-Darling test could be used for class comparison analysis; in class prediction analysis, models suitable for analyzing count data should be used (*i.e.* Negative Binomial, Poisson distribution based models).

To conclude, in this study we implemented a global strategy for the analysis of plasma miRNAs. In class comparison the combination of the results of the t- and the AD tests can be considered valuable to detect miRNAs with significant FC or more general distributional differences between classes, which could reveal hidden differential biological processes worth to be considered for building predictive tools. The use of robust miRNA selection procedure together with multivariable modeling as a strategy employed in class prediction can guarantee result generalizability and be useful to explore the interconnections among the selected miRNAs, which are essential for highlighting their inherent biological dependences.

Figures

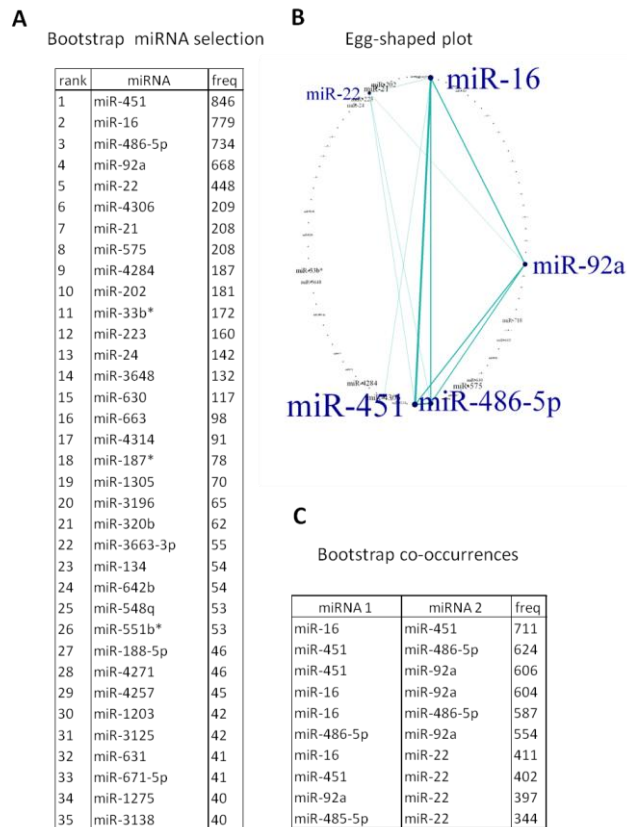


Figure 1. Results of the first step of class prediction performed in the training set raw data. A) Bootstrap occurrences of the top 35 miRNAs included in the chosen model. B) Egg-shaped plot. A filter was applied to show only the features with at least 300 co-occurrences. C) Bootstrap co-occurrences of the most interconnected miRNAs.

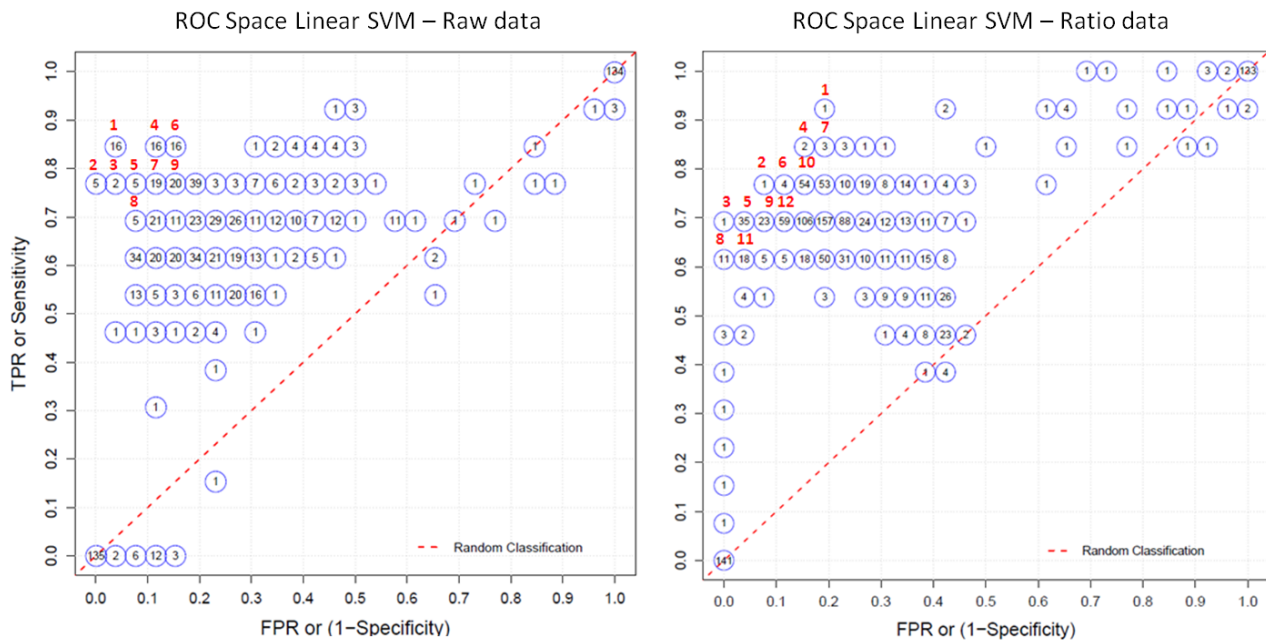


Figure 2. Results of the second step of class prediction performed in the training set raw and ratio data. 'ROC space' plot representing the classification performance of different models for class prediction in terms of false positive rate (FPR) and true positive rate (TPR) in the training set raw data (Panel A) and ratio data (Panel B). As true for the ROC curves, ideal models are those closest to the point (0,1), corresponding to 100% sensitivity and specificity.

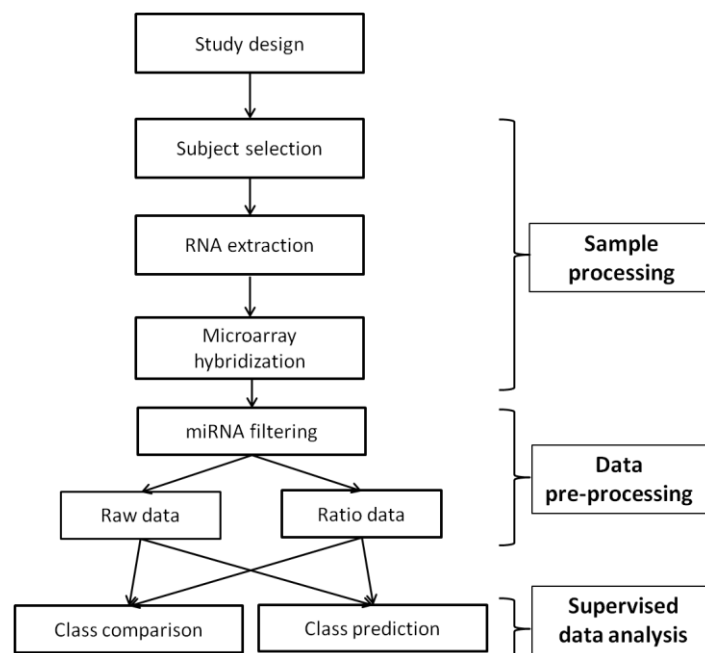
Tables

Classification performance of the best performing groups of models evaluated in the training set					Parameters of the chosen model			Classification performance of the chosen model evaluated in the testing set		
Group ID	N models	Sens	Spec	Youden	N miR	SVM Cost	SVM Weights	Sens [CI]	Spec [CI]	Youden [CI]
Raw data										
1	16	0.85	0.96	0.81	35	10	(0.5; 0.5)	0.77 [0.54-0.92]	0.77 [0.61-0.92]	0.54 [0.23-0.81]
2	5	0.77	1.00	0.77	35	1	(0.5; 0.5)	0.85 [0.61-1.00]	0.81 [0.65-0.92]	0.65 [0.38-0.85]
3	2	0.77	0.96	0.73	30	1	(0.5; 0.5)	0.85 [0.61-1.00]	0.85 [0.69-0.96]	0.69 [0.42-0.88]
4	16	0.85	0.88	0.73	40	10	(0.5; 0.5)	0.77 [0.54-0.92]	0.73 [0.54-0.88]	0.50 [0.19-0.77]
5	5	0.77	0.92	0.69	35	1	(0.4; 0.6)	0.85 [0.61-1.00]	0.73 [0.54-0.88]	0.58 [0.31-0.81]
6	16	0.85	0.85	0.69	50	10	(0.5; 0.5)	0.85 [0.61-1.00]	0.69 [0.50-0.88]	0.54 [0.27-0.81]
7	19	0.77	0.88	0.65	40	1	(0.4; 0.6)	0.77 [0.54-0.92]	0.69 [0.50-0.85]	0.46 [0.15-0.73]
8	5	0.69	0.92	0.61	3	100	(0.4; 0.6)	0.77 [0.54-0.92]	0.96 [0.88-1.00]	0.73 [0.46-0.92]
9	20	0.77	0.85	0.61	5	10	(0.4; 0.6)	0.69 [0.46-0.92]	0.92 [0.81-1.00]	0.61 [0.35-0.85]
Ratio data										
1	1	0.92	0.81	0.73	500 (88)	0.01	(0.2; 0.8)	0.92 [0.77-1.00]	0.65 [0.46-0.85]	0.58 [0.31-0.81]
2	1	0.77	0.92	0.69	17 (16)	0.01	(0.3; 0.7)	0.77 [0.54-0.92]	0.92 [0.81-1.00]	0.69 [0.42-0.92]
3	1	0.69	1.00	0.69	90 (50)	0.01	(0.5; 0.5)	0.69 [0.38-0.92]	1.00 [1.00-1.00]	0.69 [0.38-0.92]
4	2	0.85	0.85	0.69	150 (66)	0.01	(0.2; 0.8)	0.92 [0.77-1.00]	0.69 [0.50-0.85]	0.61 [0.38-0.81]
5	35	0.69	0.96	0.65	4 (5)	0.1	(0.5; 0.5)	0.77 [0.54-0.92]	1.00 [1.00-1.00]	0.77 [0.54-0.92]
6	4	0.77	0.88	0.65	500 (88)	0.01	(0.4; 0.6)	0.92 [0.77-1.00]	0.77 [0.58-0.92]	0.69 [0.46-0.88]
7	3	0.85	0.81	0.65	600 (88)	0.01	(0.2; 0.8)	0.92 [0.77-1.00]	0.65 [0.46-0.85]	0.58 [0.31-0.81]
8	11	0.61	1.00	0.61	2 (3)	0.1	(0.5; 0.5)	0.77 [0.54-0.92]	1.00 [1.00-1.00]	0.77 [0.54-0.92]
9	23	0.69	0.92	0.61	3 (4)	0.1	(0.4; 0.6)	0.77 [0.54-0.92]	0.96 [0.88-1.00]	0.73 [0.50-0.92]
10	54	0.77	0.85	0.61	4 (5)	0.1	(0.3; 0.7)	0.77 [0.54-0.92]	0.88 [0.73-1.00]	0.65 [0.38-0.88]
11	18	0.61	0.96	0.58	3 (4)	0.1	(0.5; 0.5)	0.77 [0.54-0.92]	1.00 [1.00-1.00]	0.77 [0.54-0.92]
12	59	0.69	0.88	0.58	2 (3)	10	(0.4; 0.6)	0.77 [0.54-0.92]	1.00 [1.00-1.00]	0.77 [0.54-0.92]

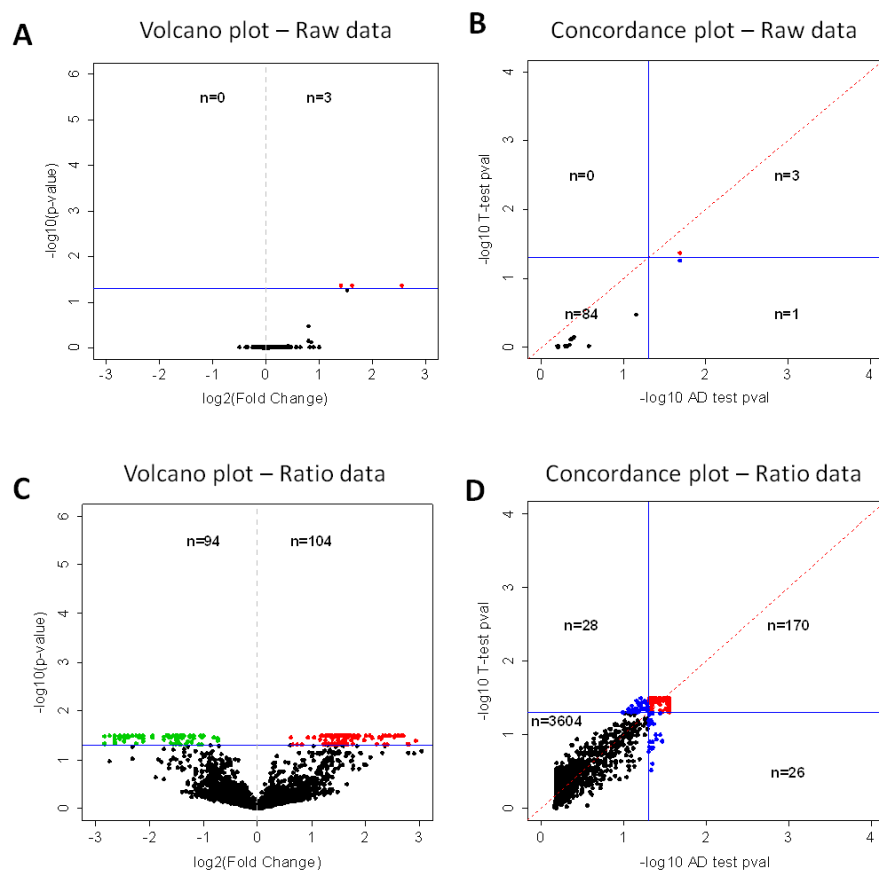
Table 1. Model classification performance measures in the training and validation sets with raw and ratio data.

Abbreviations. ‘Group ID’: ID of the groups of best performing models; ‘N models’: number of models in each group, showing a specific classification performance; ‘Sens’: sensitivity; ‘Spec’: specificity; ‘N miR’: number of miRNAs included in the model chosen in each group for containing the smallest number of miRNAs; ‘SVM cost’: cost parameter of the linear SVM model; ‘SVM weights’: weight parameter of the linear SVM. In the last three columns, testing set classification performance measures are reported together with the corresponding bootstrap 95% confidence intervals (CI).

Additional Figures



Additional figure 1. Workflow of the strategy used for sample processing, data pre-processing and supervised data analyses.



Additional figure 2. Class comparison results in the training set with raw and ratio data. T-test volcano plots and concordance plots between t- and Anderson-Darling (AD) test for raw data (panels A and B) and ratio data (panels C and D).

5.2 Secondary ElectroSpray Ionization-Mass Spectrometry data

Secondary electrospray ionization-mass spectrometry and a novel statistical bioinformatic approach identifies a cancer-related profile in exhaled breath of breast cancer patients: A pilot study

Abstract

Breath analysis represents a new frontier in medical diagnosis and a powerful tool for cancer biomarker discovery due to the recent development of analytical platforms for the detection and identification of human exhaled volatile compounds. Statistical and bioinformatic tools may represent an effective complement to the technical and instrumental enhancements needed to fully exploit clinical applications of breath analysis. Our exploratory study in a cohort of 14 breast cancer patients and 11 healthy volunteers used secondary electrospray ionization-mass spectrometry (SESI-MS) to detect a cancer-related volatile profile. SESI-MS full-scan spectra were acquired in a range of 40-350 mass-to-charge ratio (m/z), converted to matrix data and analyzed using a procedure integrating data pre-processing for quality control, and a two-step class prediction based on machine learning techniques, including a robust feature selection, and a classifier development with internal validation. MS spectra from exhaled breath showed an individual-specific breath profile and high reciprocal homogeneity among samples, with strong agreement among technical replicates, suggesting a robust responsiveness of SESI-MS. Supervised analysis of breath data identified a support vector machine (SVM) model including 8 features corresponding to m/z 106, 126, 147, 78, 148, 52, 128, 315 and able to discriminate exhaled breath from breast cancer patients from that of healthy individuals. Our data highlight the significance of SESI-MS as an analytical technique for clinical studies of breath analysis and provide evidence that our noninvasive strategy detects volatile signatures that may support existing technologies to diagnose breast cancer.

Introduction

High-throughput platforms for cancer biomarker discovery are currently focused largely on genomic and proteomic studies. A complementary approach consists in comparing the entire metabolome profile of clinical samples to detect the significant metabolic changes occurring in cancer cells [1]. Metabolomics reflects changes in phenotype and thus function, thereby representing a powerful tool in addition to genomic and proteomic-based approaches to detect cancer development [2]. Cancer metabolites can be studied in virtually all body fluids including human breath. In this context, 'breathomics' has recently been defined as the metabolomic study of exhaled air mainly focusing on the characterization of health-related volatile organic

compounds (VOCs) [3,4]. Human breath analysis-based diagnosis has unique advantages, including simplicity, safety, minimal invasiveness, painlessness and readily acceptance by patients. Once fully developed, the use of such techniques may be particularly appropriate not only in early diagnosis, but also in on- and off-line management of pediatric patients, in all medical conditions requiring frequent diagnostic assessments and in monitoring therapeutic protocols or during the surgical procedures [5,6]. In breast cancer (BC), about 60% of diagnosed invasive BCs remain localized at the time of diagnosis and the 5 year-survival is nearly 100%; however, survival rates drop to 85% and 25% if regional or distal tissue invasion occurs, respectively [7]. Noninvasive identification of molecular markers that pinpoint small lesions, invisible by imaging techniques, could greatly improve the cure rate of BC and reduce its related mortality. Indeed, reports indicate that BC can be detected by canine olfaction [8] and by gas chromatography (GC)/ mass spectrometry (MS) [9].

Interest in studies aimed at identifying clinically relevant exhaled compounds, as pioneered by Pauling et al. [10], has led to a significant development of appropriate analytical techniques for the detection and identification of human exhaled VOCs [11-14]. One such technique, secondary electrospray ionization MS (SESI-MS) [15,16], has been shown to efficiently detect trace gas-phase compounds in breath or in any other matrix in real time [17]. While this technology has recently been used to identify bacterial pathogens [18,19] and to characterize potential differences between patients with chronic obstructive pulmonary disease [20], dedicated statistical and bioinformatic tools that complement the technical and instrumental 3 enhancements in analyzing breath-derived data are still lacking [21].

Our present study, exploring the value of SESI-MS technology together with novel statistical analysis tools in cancer biomarker discovery, supports the notion that this approach can identify a cancer-related volatile signature able to discriminate exhaled breath of BC patients from that of healthy controls.

Materials and methods

Subjects

A total of 25 women participated to this study, including 14 BC patients (cases) and 11 healthy volunteers (controls). Before surgery, patients were diagnosed with BC following the standard procedure in Fondazione IRCCS Istituto Nazionale dei Tumori; none of the patients received pharmacological treatment before breath sampling. Participants were asked not to smoke, eat, drink (except water), brush their teeth or use lipstick for at least 2 h before analysis. The study was approved by the Medical Ethics Committee of Fondazione IRCCS Istituto Nazionale dei Tumori (INT 122/14).

Sample collection

Breath samples were collected on 4 different days into 2-L inert plastic bags with a valve and disposable mouthpiece (ISB, Gerenzano, Italy) previously sterilized at 40°C and 500 mTorr in the presence of H₂O₂. For 22 subjects (12 cases and 10 controls), 2 replicates were sampled within 5 minutes. To minimize variability in sample collection, storage and processing, human breath was sampled within 10 days in the same conditions for cases and controls, and plastic bags containing human breath were kept at 10°C until analysis by SESI within 2 h of collection [22] using a mass spectrometer dedicated exclusively to analysis of breath samples.

Mass spectrometry

Mass spectrometer HCT Ion Trap (Bruker Daltonics, Billerica, MA, USA) coupled to a lab-built SESI source [17] was operated in the positive ion mode. Full-scan spectra were acquired in a range of 40-350 m/z; ion source parameters were capillary 3800 V, dry gas 2 L/min and temperature 40°C. The MS instrument was slightly modified to allow admission of exhaled breath as described [22]. ES buffer (0.1% formic acid in H₂O) was infused at a flow rate of 130 nL/min by a syringe pump located outside the instrument [23].

Data acquisition and conversion to matrix

Hystar software (Bruker Daltonics, Breme, Germany) was used for data acquisition. MS spectra data of the volatile fraction were converted to the ascii xy format using Data Analysis software (Bruker Daltonics, Breme, Germany) and analyzed using the NIST database approach for pattern recognition [24]. Custom Perl (<http://www.perl.org/>) script served to optimize absolute values of each MS signal by approximating m/z and eliminating decimal places, with values then included in a [feature x sample] matrix.

Statistical and bioinformatic methods

Analyses were carried out using R software, version 2.15.2 and Bioconductor. Test results were considered significant at p-value < 0.05.

Data pre-processing

Data pre-processing (Figure 1) included five steps. Steps I, III, IV refer to analytical procedures for data quality control using statistical and bioinformatic methods imported from gene expression studies. Step I involved assessment of concordance between technical replicates using concordance and Bland-Altman plots [25] and estimation of the Lin's concordance correlation coefficient (CCC) [26] with the corresponding bootstrap 95% confidence interval [27], based on a user-defined R function, which exploits the `epi.ccc` function included in the *epiR* package. The Bland-Altman plot identified possible outliers, *i.e.*, replicates outside the

confidence bands, the mean of which could be a biased estimate of the true value. In step II, features not detected in at least 50% of samples were filtered and discarded. Step III was data normalization [28] using the quantile method (normalizeBetweenArrays function in *limma* package) to impose the same empirical feature distribution to each subject. Batch effects arising from different dates of sample collection (step IV) were corrected using the ComBat (Combating Batch Effects When Combining Batches of Gene Expression Microarray Data) method [29]. The ComBat function in the *sva* R package was used for this task. Dendrograms based on hierarchical clustering of subjects before and after Combat correction were generated to visualize the effectiveness of adjustment for batch effects. Subjects were identified according to dates of sample collection (batches); average linkage was used as the linkage criterion to construct the hierarchical cluster tree and distance was measured as one minus the Spearman correlation coefficient [30] such that 2 subjects exhibiting a strong positive correlation are closer, possibly reflecting the same feature profile between paired subjects. The *heatmap.2* function in the *gplots* package was used to perform hierarchical clustering. In step V, missing values in feature replicates were imputed after data normalization as described by Karpievitch et al. [31].

Supervised analyses

Class comparison was performed using the nonparametric Wilcoxon-Mann-Whitney test, adjusting p-values for multiple testing by the Benjamini-Hochberg method [32] to control for false-discovery rate (FDR); class prediction involved 'bootstrap feature selection' [33], followed by classifier development and its internal validation (Figure 1). In the first step, 1000 bootstrap samples were drawn from the original dataset and the features were robustly ranked according to the proportion of bootstrap samples in which they were jointly identified as independent class predictors by 3 different classification algorithms, *i.e.* prediction analysis for microarrays (PAM) [34], random forest (RF) with Boruta feature selection [35] and SVM algorithm with L_1 -penalization (L_1 SVM) [36] or, alternatively, elastic smoothly clipped absolute deviation (elastic SCAD) SVM [37]. An egg-shaped plot was initially used to summarize the bootstrap-derived feature occurrences (nodes) and co-occurrences (edge thickness). The larger the node, the more often the corresponding feature occurred in the bootstrap samples; the thicker the edge between two nodes/features, the more often they were selected together in the bootstrap samples. Bootstrap selection was performed using modified *doBS* and importance *igraph* functions in the *bootfs* package. To develop the cross-validated classifier, we applied linear SVM models, well-established machine-learning techniques used for high-dimensional data such as 'omics' data [38-39]. A linear SVM model, implemented using the function *svm* in the *e1071* package, requires the tuning of only two parameters (cost parameter and class weights). Models were fitted by varying parameters and number of included features, forwardly selected

according to the bootstrap-generated list. Each model was then internally validated using a leave-one-out cross-validation (LOOCV) procedure to optimize parameters and estimate the model classification ability by computing sensitivity, specificity and Youden index (sensitivity + specificity - 1). The false-positive rate (FPR, 1 - specificity) and true-positive rate (TPR, sensitivity) were graphically represented in the 'ROC space' plot, which can be seen as a generalization of the ROC curve representing the classification performance of the different linear SVM models. The final model used to develop the classifier was chosen based on both best classification performance, as indicated by the highest Youden index, and smallest number of features included in the model. The heatmap was generated by clustering feature values and using the 'one minus the Spearman correlation coefficient' as distance metric and the average linkage as linkage criterion.

'Feature importance analysis' was performed by generating 1000 permuted data sets and running the L_1 SVM-based bootstrap selection procedure on each random data set. The best 3 features of each selection were extracted, compared to the features bootstrap-selected on not permuted original data and the co-occurrence in permuted and original datasets were calculated.

Results

Exhaled breath from BC patients and healthy controls were sampled in duplicate within 2 hours before SESI/MS analysis. Patients' breath samples were collected 3-24 hours before surgery. Histology confirmed the presence of a tumor mass at the time of sampling. Clinical and pathological characteristics of patients revealed consistency in BC consecutive cohorts, *i.e.*, average tumor size 2 cm, 64% node-negative tumors, 71% ER (Estrogen Receptor) positive, and 71% grade I-II.

Data pre-processing

MS spectra from exhaled breath of a subject were highly similar in replicates and each participant showed an individual-specific breath profile (Figure 2A), consistent with previous studies [22,40,41]. Breath mass spectra were processed and converted to a final matrix including 351 features and 47 samples (16497 total values). About 17% of values (2838/16497) were missing and equally distributed between the first and second replicates and between cases and controls, suggesting that missing values were missing at random (MAR). In step I of data pre-processing (Figure 1), the concordance correlation coefficient (CCC, Figure 2B) for the 22 subjects with technical replicates, together with the concordance plots and the Bland-Altman plots, confirmed the agreement between technical replicates (CCC range: 0.89-0.99). Overall, variability of SESI-MS measurements was higher for signals in the low-abundance region. Based on the concordance analysis, the mean of the two replicate values for each feature in each

sample was calculated and, after filtering (step II), 296 features remained for subsequent analyses. Based on the above results and considering the quality and characteristics of SESI breath data and their matrix data structure, statistical and bioinformatics tools for data pre-processing and supervised analyses of gene expression data were applied. After quantile normalization (step III) and Combat correction (step IV), the dendrogram obtained from hierarchical clustering indicated grouping of samples independent of daily batches. Seventy values missing in both replicates and occurring in 18 samples and 30 features were imputed using the median of each feature (step V) under the MAR assumption.

Classifier development

The resulting data matrix [296 features x 25 subjects] was further statistically analyzed in a supervised setting (Figure 1) in an effort to identify signals reflecting exhaled compounds that may be valuable in identifying BC, based on the mass spectrometric fingerprints. Class comparison revealed 35 features in breath that differed significantly between cases and controls (Figure 3), 24 (69%) of which were present at higher levels in the exhaled breath from cases. This subset included the features with m/z 148 and 128, showing FDR-adjusted p -values of 0.0259 and 0.0770, respectively (nominal p -values: $9e-05$ and $5e-04$). In class prediction analysis, we first attempted to identify the most informative signals using the bootstrap feature selection strategy, based on 3 different classification algorithms: PAM, RT and L_1 SVM. Bootstrap results are represented by an egg-shaped plot (Figure 4A) that provides an immediate overview of the feature relevance in terms of bootstrap occurrence (node size) and co-occurrence (edge thickness); the latter can suggest possible structural and/or biological links among molecules. The signal detected at m/z 106, selected in 346 of 1000 bootstrap samples was the most discriminative, followed in decreasing order by signals at m/z 126 and 147 (345 of 1000), 78 (331 of 1000), 148 and 52 (322 of 1000) and 128 (259 of 1000). Figure 4B shows the frequency of bootstrap occurrences and co-occurrences of the features represented in the egg-shaped plots. The most frequent co-occurrences involved feature corresponding to m/z 126 and were jointly selected with feature 147 (202 of 1000 bootstrap samples), 148 (190 of 1000), 128 (185 of 1000) and 315 (160 of 1000); note that signal at m/z 148 co-occurred with its isotope at m/z 147 (161 of 1000). Signals selected in at least one bootstrap sample were used in a multivariable context to develop a cross-validated linear SVM classifier. The 'ROC space' in Figure 5A shows the results of the different SVM models in terms of cross-validated classification performance. The model with sensitivity = 0.93, specificity = 0.91 and Youden index = 0.84, including 8 features (m/z 106, 126, 147, 78, 148, 52, 128, 315) was used to develop the classifier, characterized by the best performance in discriminating exhaled breath from BC patients and by the smallest number of features among the models with equal discriminating performance. Figure 5B shows a heat map representing the abundance of the 8 features in each sample. Overall, supervised analysis

indicated that the mass regions, 147-148, 126-128, 106 and 315 included signals originating from molecules possibly chemically and/or functionally related and differentially over-detected in exhaled breath of BC patients.

The bootstrap feature selection was also performed by applying the elastic SCAD SVM, in conjunction with PAM and RF, obtaining similar results as those achieved using L_1 -SVM. Both feature selection algorithms clearly tends to pick isotopic features (e.g., m/z 147-148), suggesting that they efficiently selects interconnected signals.

Finally, to ensure that the discriminative features were not selected merely by chance, we calculated how often the 8 features of the final classifier were the top 3 in the bootstrap selection classifications obtained from 1000 permuted data sets, in which a random association between features and classes (cases and controls) was generated. None of the 8 features was selected as top 3 in 848 permuted data sets, one of the 8 features was top 3 in 141 selections, and a couple of features appeared jointly in 11 selections. The most predictive feature, *i.e.* 106, appeared 2% of the times as top 3; the corresponding percentages for the other features were: 0.4% for 126 and 147, 5.9% for 78, 0.1% for 148, 7.2% for 52, 0.2% for 128 and 0.1% for 315. This indicates that the 8 BC-related features were randomly bootstrap-selected in permuted datasets, supporting the robustness of the classifier.

Discussion

The recent dramatic improvement of analytical platforms for metabolic profiling has provided evidence encouraging the use of metabolic biomarkers as a valuable tool for cancer detection [1]. In the case of biomarker discovery in exhaled breath, sample handling, chemical analysis and subsequent data mining await standardization after further exploration of novel and suitable approaches. In particular, there is a well-identified need for defining data pre-processing techniques and supervised methods in breath analysis, as recently highlighted by Smolinska and coworkers [21]. Here, we explored the clinical usefulness of a combination of a promising breath analytical tools, *i.e.*, SESI-MS, and a statistical and bioinformatic tool for data pre-processing and classifier development. Using this novel strategy, we identified a BC-related signature able to classify patient and control breathprints with sensitivity and specificity above 0.9.

Because all 'omics' studies are generally affected by several sources of variability, including inherent biological variation as well as data noise due to intrinsic technical variability, there is a need for efficient governance of non-biological variability in the pre-analytical and analytical phases to minimize data misinterpretation. One of the main advantages of the analytical platform used herein resides in the rapid screening of breath metabolites made possible by SESI-MS, resulting in rich MS fingerprints without any sample preparation. Indeed, the complete SESI-MS analysis of two replicate samples collected in appropriate bags was typically accomplished in less than one minute. The accuracy of replicate profiles and high reciprocal homogeneity among

all samples suggest the robustness of SESI-MS measurements. In addition, data produced using SESI-MS analysis were high-dimensional data of quality comparable to gene expression analysis output, prompting us to use methods originally developed for gene expression data in analyzing SESI-MS breathprints. Together, the properties of SESI-MS suggest its particular suitability for large-scale clinical breath analyses.

In the post-analytical step, we used data pre-processing techniques to correct for residual systematic technical or non-biological experimental variation. The Combat correction, in particular, was used to adjust for batch effects arising from the breath sample collection procedure and analysis performed on different days.

With the aim of developing a VOC signature that accurately discriminates between cases and controls, we used a two-step procedure involving current and new approaches for data analysis and representation that promises to ensure generalization of results and provide insights into feature interconnections. Bootstrap feature selection raised a robust and specific feature ranking, as supported by the random selection of the BC-related features when the bootstrap procedure was applied to 1000 permuted datasets. The bootstrap feature selection was based on conceptually different machine-learning algorithms: PAM, RF and two alternative SVM algorithms, *i.e.* L_1 or elastic SCAD. The above algorithms were chosen because they can overcome the ‘curse of dimensionality’ typical of ‘omics’ data, *i.e.*, feature numbers much larger than subject numbers, and are representative of methodological categories using different decision rules for classification. PAM provides simplicity and interpretability, while RF and SVMs algorithms are suitable for complex classification patterns. RF is nonparametric, which is desirable especially when outliers occur, and also deals with data overfitting; however, RF only outputs importance measures, the interpretation of which is controversial in the presence of correlated features [42]. The recent elastic SCAD SVM [37] has been proposed as an effective method for considering the correlation structures in the input data (grouping effect) and applied to develop a miRNA-based classifier able to discriminate hemolyzed and not hemolyzed plasma samples [42]. The L_1 -SVM showed high prediction accuracy for models in which the sparsity assumption (small number of nonzero parameters) is tenable [36], as in the case of ‘omics’ data characterized by few predictive variables. In our application the two alternative SVM algorithms generated comparable bootstrap feature rankings, but the computation time of bootstrap process was dramatically reduced using L_1 -SVM jointed to PAM and RT, addressing this setting of our pipeline to biomarker discovery in large cohort of patients. We summarized the bootstrap feature selection in an egg-shaped plot, allowing immediate visualization of the most discriminative features and their co-occurrences and highlighting the possible interconnections underlying the structural/biological framework usually hidden in massive data. Lastly, we derived a molecular signature associated with cancer patient breath samples using a linear SVM model based on the ‘rule’ of best classification performance, as indicated by the highest Youden

index, and smallest number of features included in the model. Linear SVM models require the tuning of only two parameters but, like all SVM models, do not allow probability estimation or ROC curve generation; nevertheless, a binary classifier can be easily derived from SVM predictions, since they cluster around two different values.

While conclusions from our study are limited by our small sample size, our statistical and bioinformatic strategy for breath analysis has already been adapted to other 'omics' data, including microRNA data [42] and a complex LC-MS dataset from high-resolution Orbitrap analysis of plasma samples (Landoni, Miceli and Orlandi, in preparation). This indicates that our procedure is flexible enough to adapt the algorithm to different questions, data structure or knowledge domain.

The present study designed a general strategy effective in discovering potential volatile biomarkers to be developed for early diagnosis of cancer. The classification performance of our volatile signature awaits confirmation in larger cohorts of BC patients and non-cancer control subjects. Another important aspect is the identification of the most discriminative exhaled compounds to gain insights into the disease. This task can be undertaken by combining SESI with mass spectrometers of high resolution and fragmentation capabilities, thereby enabling unambiguous identification [44, 45].

The translation of the biomarker discovery phase to the clinical practice, beyond the validation phase, will still require a significant development of the analytical and bioinformatics procedures, tailored on the definitive classifier, to finally deliver a user-friendly tool for the rapid, simple and precise detection of cancer-related signals by breath analysis.

Conclusion

Overall, our study supports the value of SESI-MS as an analytical technique for clinical studies, since it allows rapid collection of rich metabolic breathprints, and underscores the importance of sample quality assessment and quality control of raw data from breath analysis using a robust data pre-processing techniques to address unbiased pattern discovery. Our identification of a potential cancer-related volatile signature that identifies BC patients based on their exhaled metabolic breathprint provides the foundation and rationale for further analyses aimed at developing a noninvasive diagnostic tool for prediction of BC.

Figures

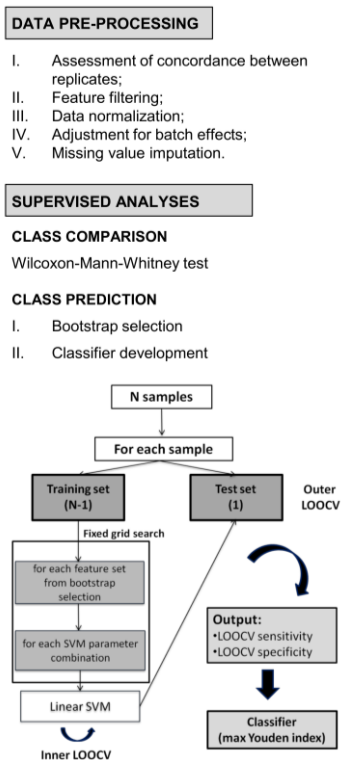


Figure 1. Workflow for data pre-processing and supervised analyses.

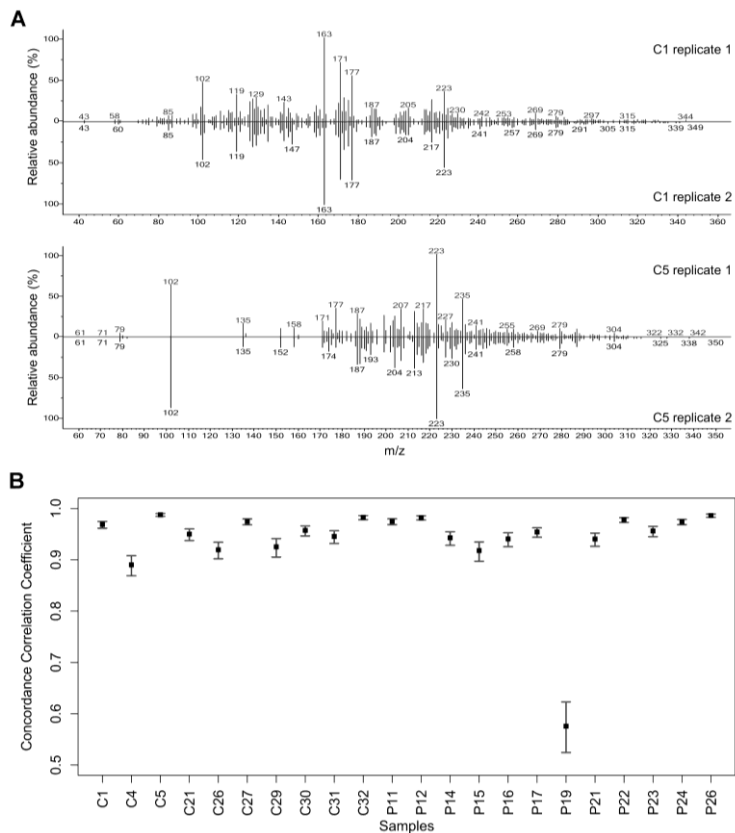


Figure 2. Quality control of pre-analytical and analytical procedures in data pre-processing. A) Evaluation of MS spectra quality based on NIST comparison of MS spectra from exhaled breath of two replicates from randomly selected subjects. B) Concordance analysis of replicates based on concordance correlation coefficient (CCC) and corresponding 95% bootstrap confidence interval (CI) for each subject.

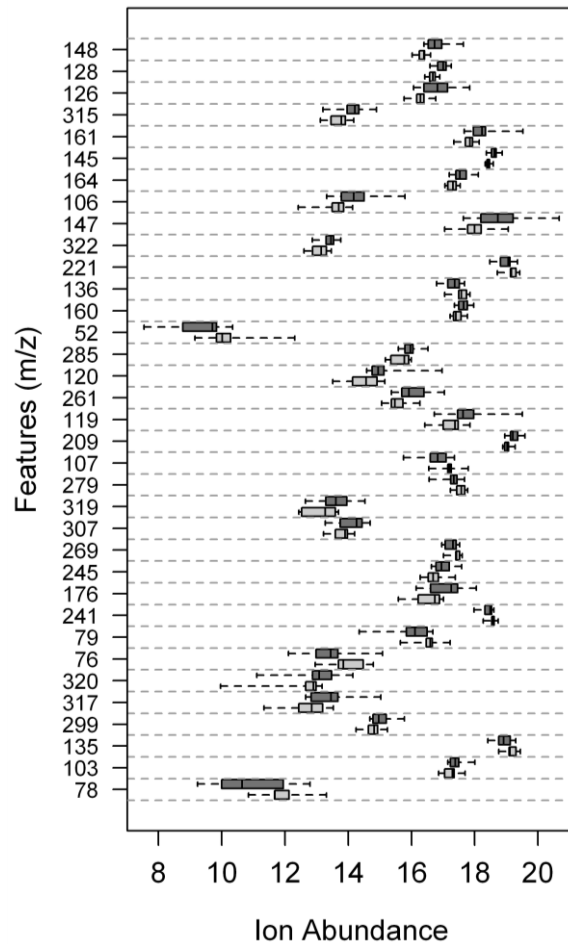


Figure 3. Class comparison analysis. Boxplots showing the distribution of the ion abundance values of the 35 significantly differentially detected features in cases (dark grey) versus controls (light grey), sorted in ascending order according to the Wilcoxon test p-value.

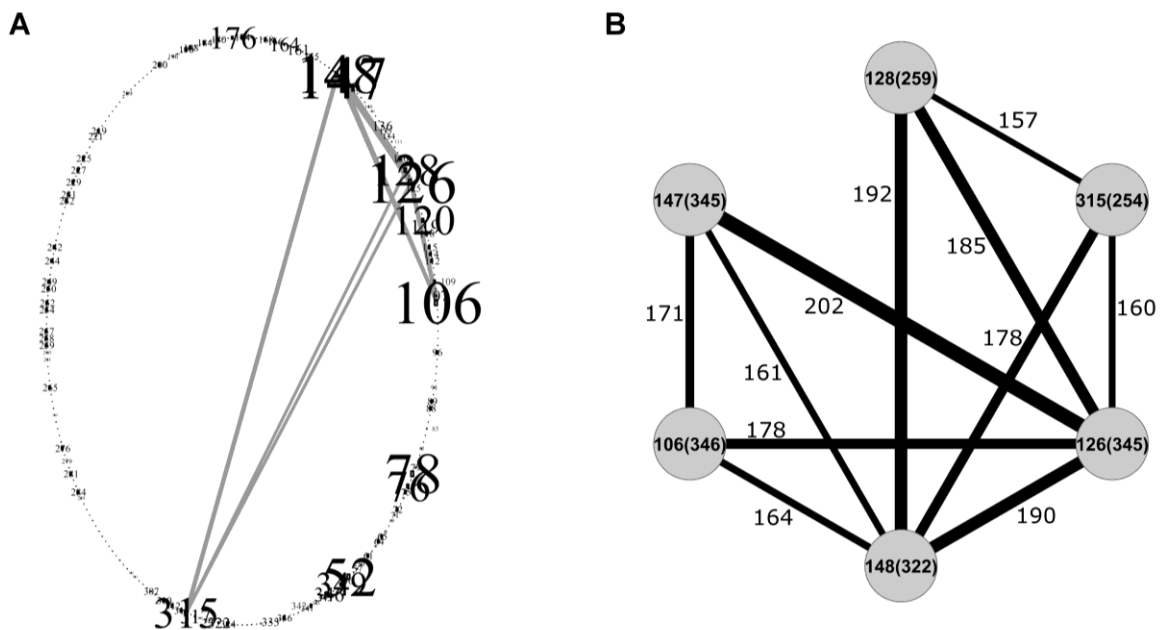


Figure 4. Bootstrap feature selection. A) Egg-shaped plot representing bootstrap results and showing features (m/z) with at least 150 co-occurrences. B) Frequency of occurrences (in brackets) and co-occurrences (next to edge) of the features represented in the egg-shaped plot.

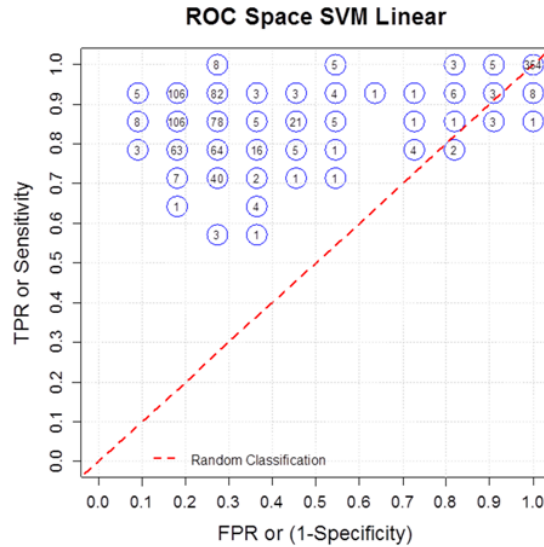


Figure 5. Classifier development. The ‘ROC space’ plot represents the LOOCV classification performance of different linear SVM models for class prediction in terms of false-positive rate (FPR; 1-specificity) and true-positive rate (TPR; sensitivity).

6. Conclusions

In the context of cancer research, we propose a pipeline for class comparison and class prediction analyses. In class comparison we recommend the use of the two-sample AD test, since our simulation study proved that it is a sensitive and flexible test able to detect general differences between feature distributions under different patterns, and thus it can keep more discriminative features than other tests for class prediction analysis. For the latter, we set up a two-step strategy able to robustly rank the features and find a parsimonious and generalizable classifier for two-class classification problems. We applied such a pipeline in different contexts of high dimensional data, adapting it on the basis of the specific data at hand. The core idea of the feature selection strategy is to jointly use three different and recently developed algorithms, each one with its pros and cons, in order to detect features that are discriminative for different aspects and thus be more prone to be validated with other data. Moreover, we paid special attention to the graphical representation of results, favoring the use of simple plots, which can be easily understandable also by non-statistician researchers. Future research will be directed to give more insights on the optimization of the Linear SVM parameters, the study of CV variants for model selection and the sensitivity of results to the use of different simulation seeds. Moreover, we aim at finding an effect measure in line with the AD test (like the FC for the t test) to be used in new volcano

plots and searching for an alternative method to the Linear SVM model for classifier implementation, with the possibility to use the Area Under the ROC Curve (AUC) as indicator of classification performance.

7. References

1. <http://www.ederaproject.it/>
2. Boeri M., Verri C., Conte D., Roz L., Modena P., Facchinetti F. et al. (2011). MicroRNA signatures in tissues and plasma predict development and prognosis of computed tomography detected lung cancer. *PNAS U.S.A* 108: 3713-3718.
3. Johnson, W. E., Li, C., and Rabinovic, A. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8 118-127.
4. Rosenbaum P.R., Rubin D.B. (1985). Constructing a control group by multivariate matched sampling methods that incorporate the propensity score. *American Statistician* 39: 33-38.
5. Lin L. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics* 45(1): 255-268.
6. R Tibshirani et al. (2002). Pre-validation and inference in microarrays, *Statist. Appl. Genet. Mol. Biol.* 1: 1-18.
7. Tobin J. (1958). Estimation of Relationships for Limited Dependent Variables. *Econometrica* 26: 24-36.
8. Landoni E. et al. Parametric and nonparametric two-sample tests for class comparison with high-dimensional 'omics' data: a simulation study. *Abstract presentation at IBC congress (Firenze)*.
9. Friedman J.H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 1189-1232.
10. Landoni E. et al. (2015). Proposal of supervised data analysis strategy of plasma miRNAs from hybridization array data with an application to assess hemolysis-related deregulation. *BMC Bioinformatics* 16: 388.
11. Koenker R. (2005). *Quantile Regression*. Cambridge University Press.
12. Fomel, S. et al. (2009). Reproducible Research. *Computing in Science & Engineering* 11(1): 5-7.
13. Troyanskaya O.G. et al. (2002). Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics* 18: 1454-61.
14. Saeys Y., Inza I., Larrañaga P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics* 23: 2507-17.
15. Pagano M. and Gauvreau K. *Principles of Biostatistics*, Duxbury/Thomson Learning, Pacific Grove, Calif., 2000.
16. Mann H. and Whitney D. (1947). On a test whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* 18: 50-60.
17. Zhang J. (2006). Powerful two-sample tests based on the likelihood ratio. *Technometrics* 48: 95-103.

18. Clarke R. (2008). Distribution of the two-sample cramer-von mises criterion for small equal samples, *Nat. Rev. Cancer* 8: 37-49.
19. Mai Q. (2013). The kolmogorov filter for variable screening in high-dimensional binary classification. *Biometrika* 100: 229-234.
20. Marozzi M. (2009). Some notes on the location-scale cucconi test. *J. Nonparametric Stat.* 21: 629-647.
21. Marozzi M. (2013). Nonparametric simultaneous tests for the location and scale testing: A comparison of several methods. *Commun. Stat-Simul C.* 42: 1298-1317.
22. Podgor M. and Gastwirth J. (1994). On non-parametric and generalised tests for the two-sample problem with location and scale change alternatives. *Stat. Med.* 13: 747-758.
23. Cucconi O. (1968). Un nuovo test non parametrico per il confronto tra due gruppi campionari. *Giornale degli Economisti* 27: 225-248.
24. Kolmogorov, A. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari* 83-91.
25. Anderson T.W. and Darling D.A. (1952). Asymptotic theory of certain 'goodness of fit' criteria based on stochastic processes. *Ann. Math. Stat.* 23: 193-212.
26. Scholz F. and Stephens M. (1987). K-sample anderson-darling tests. *J. Am. Statistic. Assoc.* 82: 918-924.
27. Conover W. *The design of simulation studies in medical statistics*. John Wiley and Sons, New York, USA, 1971.
28. Burr E. (1963). Distribution of the two-sample cramer-von mises criterion for small equal samples. *Ann. Math. Stat.* 34: 1-374.
29. Zhang J. (2002). Powerful goodness-of-fit based on likelihood ratio. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 64: 281-294.
30. Burton A., Altman D., Royston P., Holder R. (2006). The design of simulation studies in medical Statistics. *Stat. Med.* 25: 4279-92.
31. Conover W., Johnson M.E., Johnson M.M. (1981). A comparative study of tests for homogeneity of variances, with applications to the outer continental shelf bidding data. *Technometrics* 23: 351-361.
32. Marozzi M. (2011). Levene type tests for the ratio of two scales. *J. Statist. Comput. Simulation* 81: 815-826.
33. Rosenblatt M. (1952). Limiting theorems associated with variants of the von mises statistic. *Ann. Math. Stat.* 23: 1006-1016.
34. Van't Veer L., Dai H., van de Vijver M., He Y., Hart A., Mao M. et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415: 530-536.
35. Van de Vijver M., He Y.D., van 't Veer L.J., Dai H., Hart A., Voskuil D. et al. (2002). A gene expression signature as a predictor of survival in breast cancer. *NEJM* 347: 1999-2009.

36. Benjamini Y., Hochberg Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J.R. Stat. Soc. Ser. B Stat. Methodol.* 57: 289-300.
37. Veronesi U., Mariani L., Decensi A., Formelli F., Camerini T., Miceli R. et al. (2006). Fifteen-year results of a randomized phase III trial of fenretinide to prevent second breast cancer. *Ann Oncol.* 17(7): 1065-71.
38. Appierto, V. et al. (2014). A lipemia-Independent NanoDrop-based score to identify hemolysis in plasma and serum samples. *Bioanalysis* 6(9): 1215-1226.
39. <http://www.ncbi.nlm.nih.gov/geo/>
40. Efron B., Tibshirani R. (1993). *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman & Hall.
41. Tibshirani R., Hastie T., Narasimhan B., Chu G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS* 99: 6567-6572.
42. Kursa M.B., Rudnicki W.R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software* 36(11): 1-13.
43. Becker N., Toedt G., Lichter P., Benner A. (2011). Elastic scad as a novel penalization method for svm classification tasks in high-dimensional data. *BMC Bioinformatics* 12: 138.
44. Bradley P.S., Mangasarian O.L. (1998). Feature selection via concave minimization and support vector machines. *Machine Learning Proceedings of the Fifteenth International Conference* 82-90.
45. Guo Y., Graber A., McBurney R. N., Balasubramanian R. (2010). Sample size and statistical power considerations in high-dimensionality data settings: a comparative study of classification algorithms. *BMC Bioinformatics* 11: 447.
46. Tusher V., Tibshirani R., Chu C. (2001). *PNAS* 98: 5116-5121.
47. Donoho D., Johnstone I. (1994). *Biometrika* 81: 425-455.
48. Tibshirani R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58: 267-288.
49. Breiman, L., Friedman J.H., Olshen R.A., Stone C.J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.
50. Touw W.G., Bayjanov J.R., Overmars L., Backus L., Boekhorst J., Wels M., van Hijum S.A. (2012). Data mining in the Life Sciences with Random Forest: a walk in the park or lost in the jungle? *Brief Bioinformatics* 14(3): 315-326.
51. Díaz-Uriarte R., Alvarez de Andrés S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7: 3.
52. Breiman L. (2001). Random forests. *Machine learning* 45(1): 5-32.
53. Cutler A., Cutler D.R., Stevens J.R. (2012). *Random Forests*. Ensemble Machine Learning. US: Springer 157-175.

54. Breiman L. (2002). *Manual on setting up, using, and understanding random forests v3. 1*. Statistics Department University of California Berkeley, CA, USA.
55. Strobl C., Boulesteix A.L., Zeileis A., Hothorn T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics* 8(1): 25.
56. White A.P., Liu W. Z. (1994). Technical note: Bias in information-based measures in decision tree induction. *Machine Learning* 15(3): 321-329.
57. Strobl C., Boulesteix A.-L., Kneib T., Augustin T., Zeileis A. (2008). Conditional variable importance for random forests. *BMC bioinformatics* 9(1): 307.
58. Ishwaran H. (2007). Variable importance in binary regression trees and forests. *Electronic Journal of Statistics* 1: 519-537.
59. Guyon I., Elisseeff A. (2003). An introduction to variable and feature selection. *J Mach Learn Res* 3: 1157-1182.
60. Cover T.M., van Campenhout J.M. (1977). On the possible orderings of the measurement selection problem. *IEEE Transactions on Systems, Man, and Cybernetics* 7(9): 657-661.
61. Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P. et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286: 531-537.
62. Tuv E., Borisov A., Runger G., Torkkola K. (2009). Feature selection with ensembles, artificial variables, and redundancy elimination. *J Mach Learn Res* 10: 1341-1366.
63. Kursu M.B., Rudnicki W.R. (2010). Feature selection with the Boruta package. *J Stat Softw* 36(11): 1-13.
64. Deng H., Runger G. (2012). *Feature selection via regularized trees*. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN).
65. Deng H., Runger G. (2013). Gene selection with guided regularized random forest. *Pattern Recognit* 46(12): 3483-3489.
66. Boser B.E., Guyon I.M., Vapnik V.N. (1992). *A training algorithm for optimal margin classifiers*. 5th Annual ACM Workshop on COLT pp. 144-152, Pittsburgh, PA. ACM Press.
67. Vapnik V. (1995). *The Nature of Statistical Learning Theory* New York: Springer.
68. Scholkopf B., Tsuda K., Vert J.P. (2004). *Kernel Methods in Computational Biology*. MIT Press series on Computational Molecular Biology.
69. Shawe-Taylor J., Cristianini N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, MA.
70. Scholkopf B., Smola A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.
71. Duda R.O., Hart P.E., Stork D.G. (2000). *Pattern Classification*. Ed. Wiley-Interscience.
72. Guyon I., Weston J., Barnhill S., Vapnik V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning* 44(3): 438-443.
73. Kohavi R., John G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence* 273-324.

74. Hastie T., Tibshirani R., Friedman J. (2001). *The elements of statistical learning: data mining inference and prediction* New York: Springer.
75. Inza I., Sierra B., Blanco R., Larranaga P. (2002). Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems* 12: 25-33.
76. Markowitz F., Spang R. (2005). Molecular diagnosis: classification, model selection and performance evaluation. *Methods Inf Med* 44(3): 438-443.
77. Zou H., Hastie T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* 67(2): 301-320.
78. Zhang H.H., Ahn J., Lin X., Park C. (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics* 22(1): 88-95.
79. Li X., Xu R. (2008). *High Dimensional Data Analysis in Oncology* New York: Springer.
80. Fan J. (1997). Comments on 'Wavelets in Statistics: A Review' by A. Antoniadis. *J. Italian Stat. Assoc.* 6: 131-138.
81. Fan J., Li R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of American Statistical Association* 96: 1348-1360.
82. Wang L., Zhu J., Zou H. (2006). The double regularized support vector machine. *Statistica Sinica* 16: 589-615.
83. Froehlich H., Zell A. (2005). Efficient parameter selection for support vector machines in classification and regression via model-based global optimization. *Proc Int Joint Conf Neural Networks* 1431-1438.
84. Kohavi R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2: 1137-1143.
85. Cortes C., Vapnik V. (1995). Support-vector networks. *Mach Learn* 20: 273-297.
86. Stone M. (1974). Cross-validated choice and assessment of statistical predictions. *Journal of the Royal Statistical Society Series B* 36: 111-147.
87. Youden W. (1950). Index for rating diagnostic tests. *Cancer* 3: 32-35.
88. Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10(3): 61-74.
89. Scholkopf B., Smola A.J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.
90. Gonzalez L., Angulo C., Velasco F., Català A. (2006). Dual unification of bi-class Support Vector Machine formulations. *Pattern Recognition* 39(7): 1325-1332.
91. Joachims T. (2006). Training linear SVMs in linear time. *SIGKDD*.
92. Hsieh C.J., Chang K.W., Lin C.J., Keerthi S., Sundararajan S. (2008). A dual coordinate descent method for large-scale linear SVM. *Intl. Conf. on Machine Learning*.

93. Fu Z., Robles-Kelly A., Zhou J. (2010). Mixing linear SVMs for nonlinear classification. *IEEE Trans Neural Netw.* 21(12): 1963-75.
94. Seymour G. (1993). *Predictive Inference*. New York: Chapman and Hall.
95. Kohavi R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2(12): 1137-1143.
96. Devijver, P.A., Kittler J. (1982). *Pattern Recognition: A Statistical Approach*. London, GB: Prentice-Hall.
97. Larson S. (1931). The shrinkage of the coefficient of multiple correlation. *J. Educat. Psychol.* 22: 45-55.
98. Mosteller F., Wallace D.L. (1963). Inference in an authorship problem. *J. Am. Stat. Assoc.* 58: 275-309.
99. Mosteller F., Turkey J.W. (1968). *Data analysis, including statistics*. Handbook of Social Psychology. Addison-Wesley, Reading, MA.
100. Geisser S. (1975). The predictive sample reuse method with applications. *J. Am. Stat. Assoc.* 70(350): 320-328.
101. Grossman R., Seni G., Elder J., Agarwal N., Liu H. (2010). *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan & Claypool.
102. Efron B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Am. Stat. Assoc.* 78: 316-331.
103. <http://www.top500.org/lists/>
104. Martinez-Lozano Sinues P., Landoni E., Miceli R., Dibari V.F., Dugo M., Agresti R. et al. (2015). Secondary electrospray ionization-mass spectrometry and a novel statistical bioinformatic approach identifies a cancer-related profile in exhaled breath of breast cancer patients: a pilot study. *J Breath Res.* 9(3): 031001.

7.1 References for plasmatic microRNA data

1. <http://www.mirbase.org/>
2. Iorio, M., Croce, C.: MicroRNAs in cancer: small molecules with a huge impact. *J Clin Oncol* 27, 5848-5856 (2009)
3. Gandellini, P., Profumo, V., Folini, M., Zaffaroni, N.: MicroRNAs as new therapeutic targets and tools in cancer. *Expert Opin Ther Targets* 15, 265-279 (2011)
4. De Cecco, L., Dugo, M., Canevari, S., Daidone, M., Callari, M.: Measuring microRNA expression levels in oncology: from samples to data analysis. *Crit Rev Oncog* 18, 273-287 (2013)
5. Cortez, M., Calin, G.: MicroRNA identification in plasma and serum: a new tool to diagnose and monitor diseases. *Expert Opin Biol Ther* 9, 703-711 (2009)

6. Cortez, M., Bueso-Ramos, C., Ferdin, J., Lopez-Berestein, G., Sood, A., Calin, G.: MicroRNAs in body fluids-the mix of hormones and biomarkers. *Clin Oncol* 8, 467-477 (2011)
7. Arroyo, J., Chevillet, J., Kroh, E., Ruf, I., Pritchard, C., Gibson, D., et al.: Argonaute2 complexes carry a population of circulating microRNAs independent of vesicles in human plasma. *Proc Natl Acad Sci U S A* 108, 5003-5008 (2011)
8. Vickers, K., Palmisano, B., Shoucri, B., Shamburek, R., Remaley, A.: MicroRNAs are transported in plasma and delivered to recipient cells by high-density lipoproteins. *Nat Cell Biol* 13, 423-433 (2011)
9. Mitchell, P., Parkin, R., Kroh, E., Fritz, B., Wyman, S., Pogossova-Agadjanyan, E., et al.: Circulating microRNAs as stable blood-based markers for cancer detection. *Proc Natl Acad Sci U S A* 105, 10513-10518 (2008)
10. Ng, E., Chong, W., Jin, H., Lam, E., Shin, V., Yu, J., et al.: Differential expression of microRNAs in plasma of patients with colorectal cancer: a potential marker for colorectal cancer screening. *Gut* 58, 1375-1381 (2009)
11. Allegra, A., Alonci, A., Campo, S., Penna, G., Petrungaro, A., Gerace, D., et al.: Circulating microRNAs: new biomarkers in diagnosis, prognosis and treatment of cancer (review). *Int J Oncol* 41, 1897-1912 (2012)
12. Schwarzenbach, H., Nishida, N., Calin, G., Pantel, K.: Circulating microRNAs: new biomarkers in diagnosis, prognosis and treatment of cancer (review). *Nat Rev Clin Oncol* 11, 145-156 (2014)
13. Tiberio, P., Callari, M., Angeloni, V., Daidone, M., Appierto, V.: Challenges in using circulating miRNAs as Cancer Biomarkers. Article ID 731479 in press
14. Fortunato, O., Boeri, M., Verri, C., Conte, D., Mensah, M., Suatoni, P., et al.: Assessment of circulating microRNAs in plasma of lung cancer patients. *Molecules* 19, 3038-3054 (2014)
15. Leidner, R., Li, L., Thompson, C.: Dampening enthusiasm for circulating microRNA in breast cancer. *PLoS One* 8, 57841 (2013)
16. Scholz, F., Stephens, M.: K-sample Anderson-Darling tests. *Journal of the American Statistical Association* 82, 918-924 (1987)
17. Veronesi, U., Mariani, L., Decensi, A., Formelli, F., Camerini, T., Miceli, R., et al.: Fifteen-year results of a randomized phase iii trial of fenretinide to prevent second breast cancer. *Ann Oncol* 17, 1065-1071 (2006)
18. Kozomara, A. and Griffiths-Jones, S.: MiRBase: annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Research* 42, D68-D73 (2014).
19. Callari, M., Tiberio, P., De Cecco, L., Cavadini, E., Dugo, M., Ghimenti, C., et al.: Feasibility of circulating miRNA microarray analysis from archival plasma samples. *Anal Biochem* 437, 123-125 (2013)
20. <http://www.bioconductor.org/>
21. Appierto, V., Callari, M., Cavadini, E., Morelli, D., Daidone, M., Tiberio, P.: A lipemia-independent nanodrop(®)-based score to identify hemolysis in plasma and serum samples. *Bioanalysis* 6, 1215-1226 (2014)
22. Rosenbaum, P., Donald, B.: Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician* 39, 33-38 (1985)
23. Austin, P.: Optimal caliper widths for propensity-score matching when estimating differences in means and differences in proportions in observational studies. *Pharm Stat* 10, 150-161 (2011)

24. Kroh, E.M., Parkin R.K., Mitchell P.S., Tewari M.: Analysis of circulating microRNA biomarkers in plasma and serum using quantitative reverse transcription-PCR (qRT-PCR). *Methods* 50, 298-301 (2010)
25. Willems, M., Moshage, H., Nevens, F., Fevery, J., Yap, S.H.: Plasma collected from heparinized blood is not suitable for HCV-RNA detection by conventional RT-PCR assay. *Journal of Virological Methods* 42, 127-130 (1993)
26. García, M.E., Blanco, J.L., Caballero, J., and Gargallo-Viola, D.: Anticoagulants interfere with PCR used to diagnose invasive aspergillosis. *Journal of Clinical Microbiology* 40, 1567-1568 (2002)
27. Kim, D.J., Linnstaedt, S., Palma, J., Park, J.C., Ntrivalas, E., Kwak-Kim, J.Y. et al.: Plasma components affect accuracy of circulating cancer-related microRNA quantitation. *The Journal of Molecular Diagnostics* 14, 71-80 (2012)
28. Tiberio, P., De Cecco, L., Callari, M., Cavadini, E., Daidone, M., Appierto, V., et al.: MicroRNA detection in plasma samples: how to treat heparinized plasma. *J Mol Diagn* 15, 138-139 (2013)
29. <http://www.ncbi.nlm.nih.gov/geo/>
30. Boeri, M., Verri, C., Conte, D., Roz, L., Modena, P., Facchinetti, F., et al.: MicroRNA signatures in tissues and plasma predict development and prognosis of computed tomography detected lung cancer. *Proc Natl Acad Sci U S A* 108, 3713-3718 (2011)
31. Stephens, M.: Edf statistics for goodness of fit and some comparisons. *J Amer Stat Assoc* 69, 730-737 (1974).
32. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B* 57, 1289-1300 (1995)
33. Austin, P., Tu, J.: Automated variable selection methods for logistic regression produced unstable models for predicting acute myocardial infarction mortality. *J Clin Epidemiol* 57, 1138-1146 (2004)
34. Efron, B.: *An Introduction to the Bootstrap*. Chapman and Hall, Boca Raton, FL (1993)
35. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci U S A* 99, 6567-6572 (2002)
36. Kursa, M., Rudnicki, W.: Feature selection with the Boruta package. *Journal of Statistical Software* 36, 13 (2010)
37. Becker, N., Toedt, G., Lichter, P., Benner, A.: Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC Bioinformatics* 12, 138 (2011)
38. Neville P.G.: Controversy of Variable Importance in Random Forests. *Journal of Unified Statistical Techniques* 1, 15-20 (2013)
39. Cortes, C., Vapnik, V.: Support-vector networks. *Mach Learn* 20, 273-297 (1995)
40. Stone, M.: Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society Series B* 36, 111-147 (1974)
41. Youden, W.: Index for rating diagnostic tests. *Cancer* 3, 32-35 (1950)
42. Rao, Y., Lee, Y., Jarjoura, D., Ruppert, A., Liu, C., Hsu, J., et al.: A comparison of normalization techniques for microRNA microarray data. *Stat Appl Genet Mol Biol* 7, 22 (2008)

43. Wu, D., Hu, Y., Tong, S., Williams, B., Smyth, G., Gantier, M.: The use of miRNA microarrays for the analysis of cancer samples with global miRNA decrease. *RNA* 19, 876-888 (2013)
44. Choong, M., Yang, H., McNiece, I.: MicroRNA expression profiling during human cord blood-derived cd34 cell erythropoiesis. *Exp Hematol* 35, 551-564 (2007)
45. MacLellan, S., MacAulay, C., Lam, S., Garnis, C.: Pre-profiling factors influencing serum microRNA levels. *BMC Clin Pathol* 14, 27 (2014)

7.2 References for Secondary ElectroSpray Ionization-Mass Spectrometry data

1. Sreekumar, A. et al. 2009. Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression. *Nature*. 457 910-914.
2. Nicholson, J. K. and Lindon, J. C. 2008. Systems biology: Metabonomics. *Nature*. 455 1054-1056.
3. Rattray, N. J., Hamrang, Z., Trivedi, D. K., Goodacre, R., and Fowler, S. J. 2014. Taking your breath away: metabolomics breathes life in to personalized medicine. *Trends.Biotechnol.* 32 538-548.
4. Haick, H., Broza, Y. Y., Mochalski, P., Ruzsanyi, V., and Amann, A. 2014. Assessment, origin, and implementation of breath volatile cancer markers. *Chem.Soc.Rev.* 43 1423-10 1449 11
5. Amann, A and Smith, D. 2013. *Volatile Biomarkers*, 1st Edition.
6. Martinez-Lozano, Sinues P., Zenobi, R., and Kohler, M. 2013. Analysis of the exhalome: a diagnostic tool of the future. *Chest*. 144 746-749.
7. National Cancer Institute 2015. *Cancer Statistics, SEER Stat Fact Sheets: Breast Cancer*. <http://seer.cancer.gov/statfacts/html/breast.html>.
8. McCulloch, M., Jezierski, T., Broffman, M., Hubbard, A., Turner, K., and Janecki, T. 2006. Diagnostic accuracy of canine scent detection in early- and late-stage lung and 18 breast cancers. *Integr.Cancer Ther.* 5 30-39.
9. Phillips, M. et al. 2014. Rapid point-of-care breath test for biomarkers of breast cancer and abnormal mammograms. *PLoS.One.* 9 e90226.
10. Pauling, L., Robinson, A. B., Teranishi, R., and Cary, P. 1971. Quantitative analysis of 22 urine vapor and breath by gas-liquid partition chromatography. *Proc.Natl.Acad.Sci U.S.A.* 68 2374-2376.
11. Lovett, A. M., Reid, N. M., Buckley, J. A., French, J. B., and Cameron, D. M. 1979. Real-time analysis of breath using an atmospheric pressure ionization mass spectrometer. *Biomed.Mass.Spectrom.* 6 91-97.
12. Benolt, F. M., Davidson, W. R., Lovett, A. M., Nacson, S., and Ngo, A. 1983. Breath analysis by atmospheric pressure ionization mass spectrometry. *Anal.Chem.* 55 805-807.
13. Smith, D. and Spanel, P. 2005. Selected ion flow tube mass spectrometry (SIFT-MS) for on-line trace gas analysis. *Mass.Spectrom.Rev.* 24 661-700.
14. Lindinger, W. and Jordan, A. 1998. Proton-transfer-reaction mass spectrometry (PTR-8 MS): on-line monitoring of volatile organic compounds at pptv levels. *Chem.Soc.Rev.* 27 9 347-375.

15. Dillon, L. A., Stone, V. N., Croasdell, L. A., Fielden, P. R., Goddard, N. J., and Thomas, C. L. 2010. Optimisation of secondary electrospray ionisation (SESI) for the trace determination of gas-phase volatile organic compounds. *Analyst*. 135 306-314.
16. Reynolds, J. C. et al. 2010. Detection of volatile organic compounds in breath using thermal desorption electrospray ionization-ion mobility-mass spectrometry. *Anal.Chem.* 82 2139-2144 16
17. Martinez-Lozano, P. and Fernandez de la, Mora J. 2008. Direct analysis of fatty acid vapors in breath by electrospray ionization and atmospheric pressure ionization-mass spectrometry. *Anal.Chem.* 80 8210-8215.
18. Ballabio, C., Cristoni, S., Puccio, G., Kohler, M., Sala, M. R., Brambilla, P., and Martinez-Lozano, Sinues P. 2014. Rapid identification of bacteria in blood cultures by mass-spectrometric analysis of volatiles. *J.Clin.Pathol.* 67 743-746.
19. Zhu, J., Bean, H. D., Jimenez-Diaz, J., and Hill, J. E. 2013. Secondary electrospray ionization-mass spectrometry (SESI-MS) breathprinting of multiple bacterial lung pathogens, a mouse model study. *J.Appl.Physiol.*(1985). 114 1544-1549.
20. Martinez-Lozano, Sinues P., Meier, L., Berchtold, C., Ivanov, M., Sievi, N., Camen, G., Kohler, M., and Zenobi, R. 2014. Breath analysis in real time by mass spectrometry in chronic obstructive pulmonary disease. *Respiration.* 87 301-310.
21. Smolinska, A., Hauschild, A. C., Fijten, R. R., Dallinga, J. W., Baumbach, J., and van Schooten, F. J. 2014. Current breathomics--a review on data pre-processing techniques and machine learning in metabolomics breath analysis. *J.Breath.Res.* 8 027105.
22. Martinez-Lozano, P., Zingaro, L., Finiguerra, A., and Cristoni, S. 2011. Secondary electrospray ionization-mass spectrometry: breath study on a control group. *J.Breath.Res.* 5 016002.
23. Martinez-Lozano Sinues, P., Criado, E., and Vidal, G. 2012. Mechanistic study on the ionization of trace gases by an electrospray plume. *International Journal of Mass Spectrometry.* 313 21-29.
24. Sinues, P. M., onso-Salces, R. M., Zingaro, L., Finiguerra, A., Holland, M. V., Guillou, C. and Cristoni, S. 2012. Mass spectrometry fingerprinting coupled to National Institute of Standards and Technology Mass Spectral search algorithm for pattern recognition. *Anal.Chim.Acta.* 755 28-36.
25. Bland, J. M. and Altman, D. G. 1986. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet.* 1 307-310.
26. Lin, L. I. 1989. A concordance correlation coefficient to evaluate reproducibility. *Biometrics.* 45 255-268.
27. Efron, B. 1979. Bootstrap methods: Another look at jackknife. *The Annals of Statistics.* 7 1-26.
28. Callister, S. J., Barry, R. C., Adkins, J. N., Johnson, E. T., Qian, W. J., Webb-Robertson, B. J., Smith, R. D., and Lipton, M. S. 2006. Normalization approaches for removing systematic biases associated with mass spectrometry and label-free proteomics. *J.Proteome.Res.* 5 277-286.
29. Johnson, W. E., Li, C., and Rabinovic, A. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics.* 8 118-127.
30. Key, M. 2012. A tutorial in displaying mass spectrometry-based proteomic data using heat maps. *BMC Bioinformatics.* 13 Suppl 16 S10.

31. Karpievitch, Y. V., Dabney, A. R., and Smith, R. D. 2012. Normalization and missing value imputation for label-free LC-MS analysis. *BMC Bioinformatics*. 13 Suppl 16 S5.
32. Benjamini, Y. and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*. 57 8 289-300.
33. Austin, P. C. and Tu, J. V. 2004. Bootstrap methods for developing predictive models. *The American Statistician*. 58 131-137.
34. Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc.Natl.Acad.Sci U.S.A.* 99 13 6567-6572.
35. Kursa, M. B. and Rudnicki, W. R. 2010. Feature selection with the Boruta Package. *Journal of Statistical Software*. 36 1-13.
36. Hastie, T., Tibshirani, R. and Wainwright, M. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 1st Edition.
37. Becker, N., Toedt, G., Lichter, P., and Benner, A. 2011. Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC Bioinformatics*. 12 138.
38. Cortes, C. and Vapnik, V. 1995. Support-vector networks. *Machine Learning*. 20 273-297.
39. Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. 2000. Support vector machine classification and validation of cancer tissue samples 25 using microarray expression data. *Bioinformatics*. 16 906-914.
40. Martinez-Lozano Sinues, P., Kohler, M., and Zenobi, R. 2013. Human breath analysis may support the existence of individual metabolic phenotypes. *PLoS.One*. 8 e59909.
41. Wang, X. R., Lizier, J. T., Berna, A. Z., Bravo, F. G. and Trowell, S. C. 2015. Human breath-print identification by E-nose, using information-theoretic feature selection prior to classification *Sensors and Actuators, B: Chemical* 217 165-74.
42. Grömping, U. 2009. Variable importance assessment in regression: Linear regression versus random forest. *Am Stat*. 63 308-319.
43. Landoni, L., Miceli, R., Callari, M., Tiberio, P., Appierto, V., Angeloni, V. et al. 2015. Proposal of supervised data analysis strategy of plasma miRNAs from hybridisation array data with an application to assess hemolysis-related deregulation. Accepted, *BMC Bioinformatics*.
44. García-Gómez, D., Bregy, L., Barrios-Collado, C., Vidal-de-Miguel, G. and Zenobi, R. 2015. Real-Time High-Resolution Tandem Mass Spectrometry Identifies Furan Derivatives in Exhaled Breath *Anal. Chem*. 87 6919-24.
45. García-Gómez, D., Martínez-Lozano Sinues, P., Barrios-Collado, C., Vidal-De-Miguel, G., Gaugg, M. and Zenobi, R. 2015 Identification of 2-alkenals, 4-hydroxy-2-alkenals, and 4-hydroxy-2,6-alkadienals in exhaled breath condensate by UHPLC-HRMS and in breath by real-time HRMS *Anal. Chem*. 87 3087-93.

Appendix 1:R codes for Cramer-von-Mises - Podgor-Gastwirth PG2 - Cucconi - Zhang tests

User-defined codes of the implemented tests: CvM test, PG2 test, Cucconi test (asymptotic version), Cucconi test (empirical version), Zhang tests.

❖ CvM test

```
f.cvm.burr <- function(x1,x2){
vect <- sort(c(x1,x2))
binEdges = c(min(vect-1),vect,max(vect)+1)
binCounts1 = hist (x1 , breaks=binEdges, plot = FALSE)$counts
binCounts2 = hist (x2 , breaks=binEdges, plot = FALSE)$counts
sumCounts1 = cumsum(binCounts1)/sum(binCounts1)
sumCounts2 = cumsum(binCounts2)/sum(binCounts2)
sampleCDF1 = sumCounts1[1:length(sumCounts1)-1]
sampleCDF2 = sumCounts2[1:length(sumCounts2)-1]
N1=length(x1);N2=length(x2);N=N1+N2
CMstastic = (N1*N2)/(N^2)*(sum((sampleCDF1 - sampleCDF2)^2))
pValue <- 10^(-0.458-0.444*log10(CMstastic)-2.151*CMstastic)
return(list(stat=CMstastic, p.val=pValue))
}
```

❖ PG2 test

```
f.pg2 <- function(x, y){
Y = c(rep(0, length(x)), rep(1, length(y)))
R = rank(c(x,y))
pg = summary(lm(Y ~ R + I(R^2)))
pval.pg2<-1-pf(pg$fstatistic[1], pg$fstatistic[2], pg$fstatistic[3])
names(pval.pg2)<-NULL
return(pval.pg2)
}
```

❖ Cucconi test (asymptotic version)

```
f.cucconi <- function(x, y){
n1 <- length(x); n2 <- length(y); n <- n1+n2
R = rank(c(x,y))
term1=(n+1)*(2*n+1)
term2=(8*n+11)
numU=6*sum((R[1:n1])^2)-n1*term1
```

```

denU=sqrt (n1*n2*term1*term2/5)
U=numU/denU
numV = 6*sum(((n+1-R)[1:n1])^2)-n1*term1
denV=sqrt (n1*n2*term1*term2/5)
V=numV/denV
ro = (2*(n^2-4))/((2*n+1)*term2)-1
C = (U^2 + V^2 - 2*ro*U*V)/(2*(1-ro^2))
C.pval=exp(-C)
return(list(stat=C,p.val=C.pval))
}

```

❖ Cucconi test (empirical version)

```

f.cucconi.exact <- function(x,y,N=2000) {
n1 <- length(x); n2 <- length(y); n <- n1+n2
Sc <-0
Rc = rank(c(x,y))
term1=(n+1)*(2*n+1)
term2=(8*n+11)
numU=6*sum((Rc[1:n1])^2)-n1*term1
denU=sqrt (n1*n2*term1*term2/5)
U=numU/denU
numV = 6*sum(((n+1-Rc)[1:n1])^2)-n1*term1
denV=sqrt (n1*n2*term1*term2/5)
V=numV/denV
ro = (2*(n^2-4))/((2*n+1)*term2)-1
C = (U^2 + V^2 - 2*ro*U*V)/(2*(1-ro^2))
for (j in 1:N) {
R <- sample(n)
term1=(n+1)*(2*n+1)
term2=(8*n+11)
numU=6*sum((R[1:n1])^2)-n1*term1
denU=sqrt (n1*n2*term1*term2/5)
U=numU/denU
numV = 6*sum(((n+1-R)[1:n1])^2)-n1*term1
denV=sqrt (n1*n2*term1*term2/5)
V=numV/denV
ro = (2*(n^2-4))/((2*n+1)*term2)-1
cc = (U^2 + V^2 - 2*ro*U*V)/(2*(1-ro^2))
Sc = Sc + (cc > C)
}
C.pval <- Sc/N

```

```
return(list(C=C,p.C=C.pval))
}
```

❖ Zhang tests

```
f.zhang.tests <- function(x,y,N=2000) {
n1 <- length(x); n2 <- length(y); n <- n1+n2
Sc <-0; Sa<-0; Sk<-0
Rc <- rank(c(x,y))
Ra <- ceiling(rank(c(x,y)))
Rk <- ceiling(rank(c(x,y)))
```

✚ f Z_c test

```
gc <- function(m, r, M) {sum(log(m/(1:m-.5)-1)*log(M/(r-.5)-1))}
```

✚ f Z_A test

```
ga <- function (m, r, M) {d <- sort(r); D <- c(1,d,M+1)
p <- rep(0:m, D[2: (m+2)]-D[1: (m+1)] )
p[d] <- p[d]-.5; p <- p/m
m*(p*log(p+.0000000001)+(1-p)*log(1-p+.0000000001))}
wa <- (1:n-.5)*(n:1-.5)
```

✚ f Z_k test

```
P <- (1:n-.5)/n
wk <- n*(P*log(P)+(1-P)*log(1-P))
gk <- function (m, r, M) {d <- sort(r); D <- c(1,d,M+1)
p <- rep(0:m, D[2: (m+2)]-D[1: (m+1)] )
p[d] <- p[d]-.5; p <- p/m
m*(p*log(p+.0000000001)+(1-p)*log(1-p+.0000000001))}
Zc <- (gc(n1, sort(Rc[1:n1]),n) + gc(n2, sort(Rc[(n1 + 1):n]),n))/n
Za <- -sum((ga(n1,Ra[1:n1],n)+ga(n2,Ra[(n1+1):n],n))/wa)
Zk <- max(gk(n1, Rk[1:n1] , n) + gk(n2, Rk [ (n1 + 1) :n] , n) - wk)
for (j in 1:N) {
R <- sample(n)
zc <- (gc(n1,sort(R[1:n1]),n)+gc(n2, sort(R[(n1 + 1):n]),n))/n
za <- -sum((ga(n1,R[1:n1],n)+ga(n2, R[(n1+1):n],n))/wa)
zk <- max(gk(n1, R[1:n1] , n) + gk(n2, R [ (n1 + 1) :n] , n) - wk)
Sc = Sc + (zc < Zc)
Sa = Sa + (za < Za)
Sk = Sk + (zk > Zk)
}
Zc.pval <- Sc/N
```

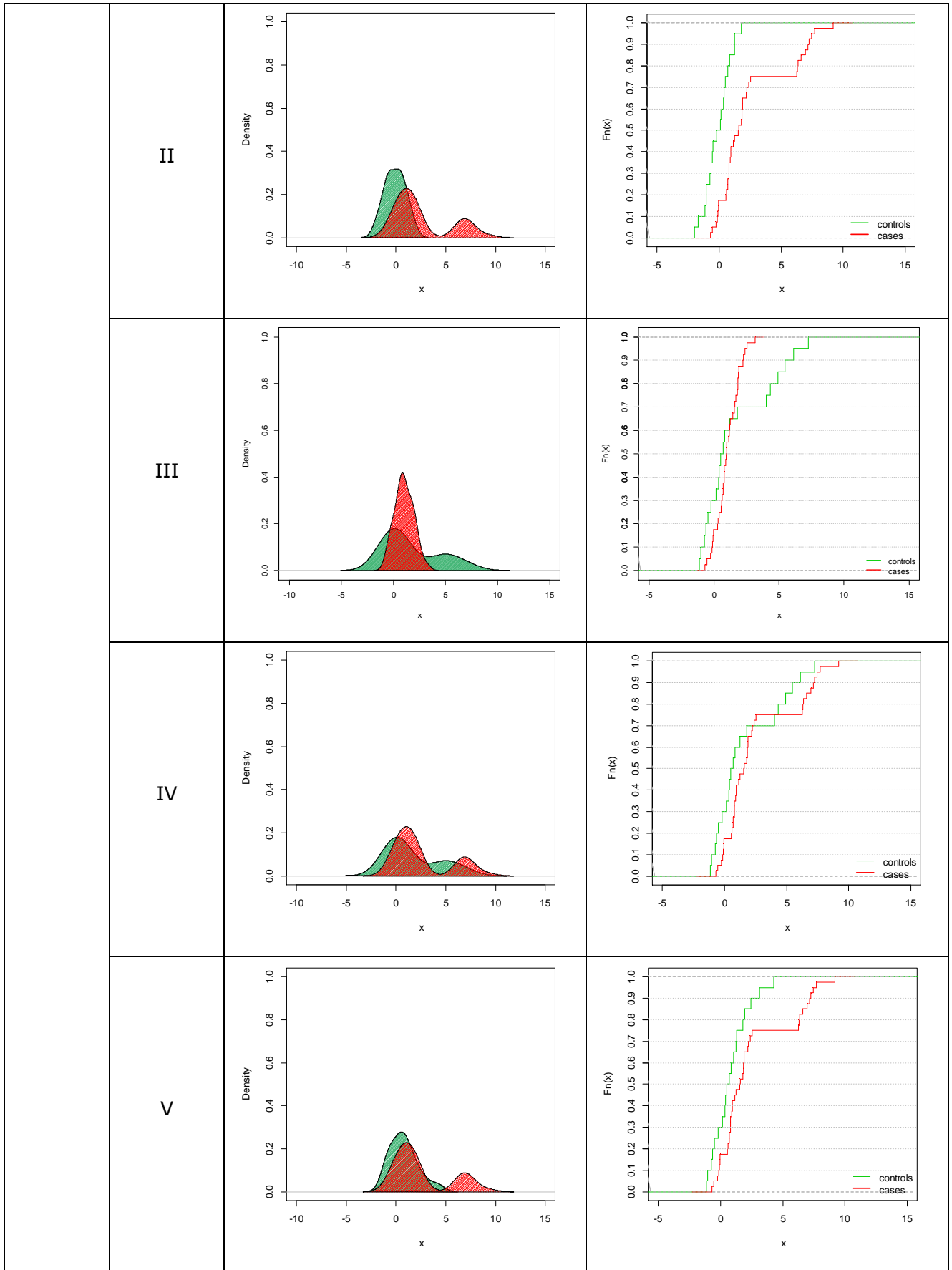
```
Za.pval <- Sa/N  
Zk.pval <- Sk/N  
return(list(Zc=Zc,p.Zc=Zc.pval,Za=Za,p.Za=Za.pval,Zk=Zk,p.Zk=Zk.pval))  
}
```

Appendix 2: Density distributions and Empirical Cumulative Distribution Functions of the simulation patterns I-VI

A2.1 Density distributions and ECDFs for $\lambda=0.80$

(m,n)	Pattern	Density distributions	ECDFs
(20,20)	I		
	II		
	III		

	IV		
	V		
	VI		
(20,40)	I		



	VI		
(40,20)	I		
	II		
	III		

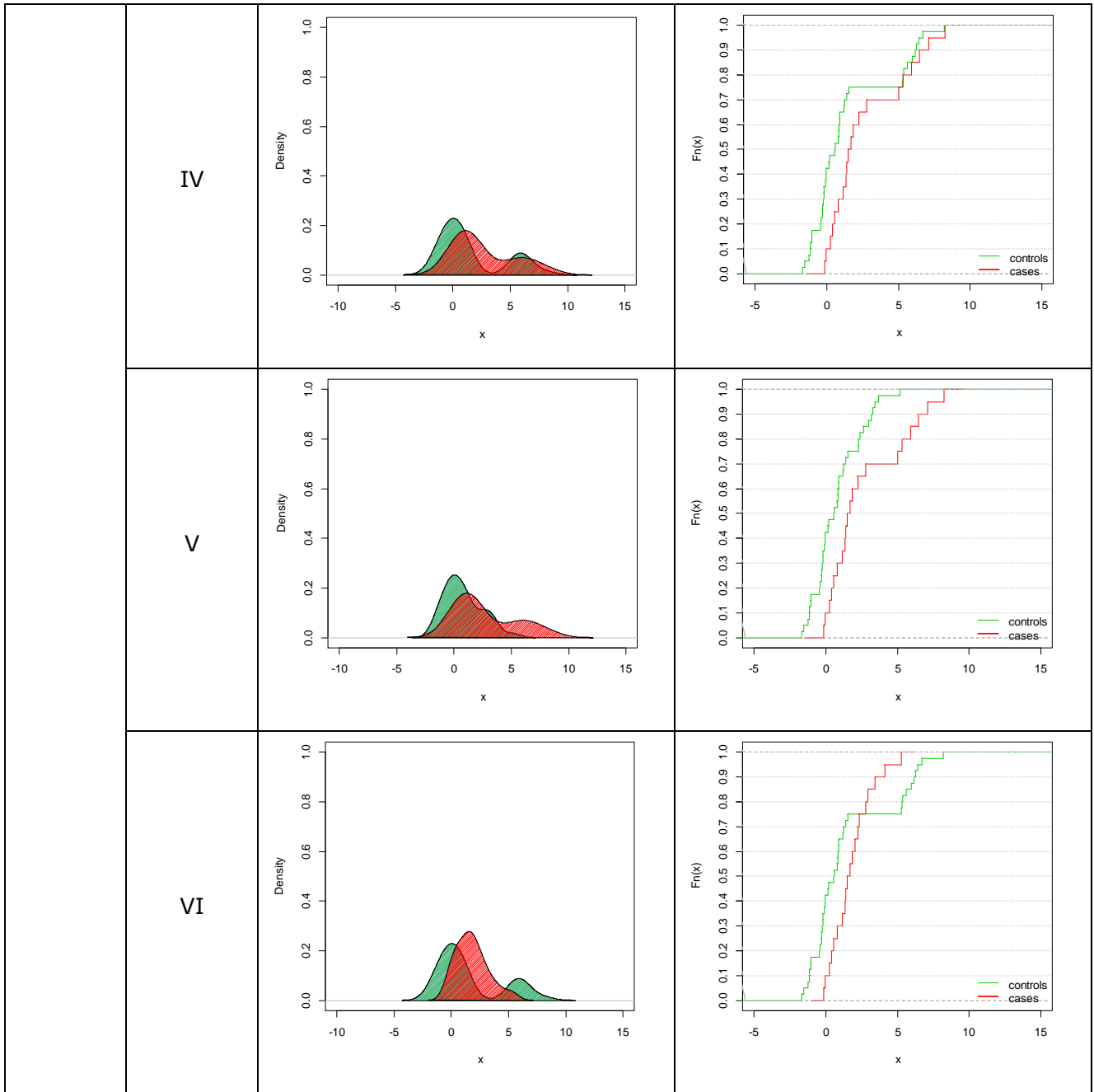
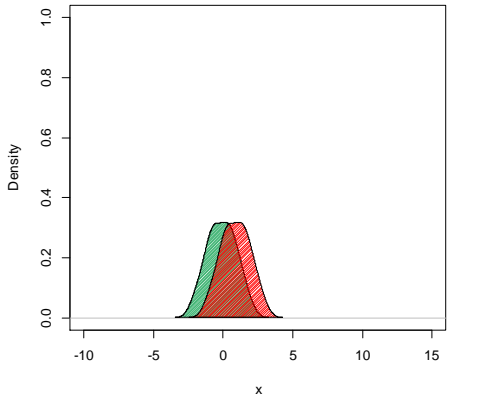
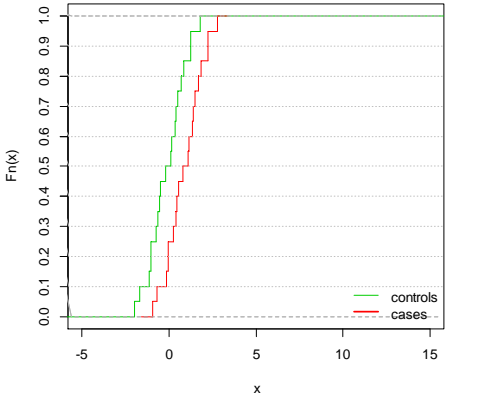
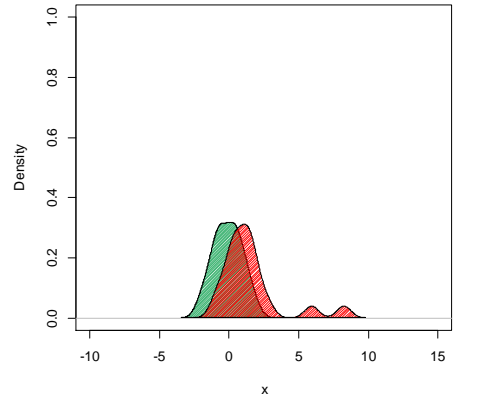
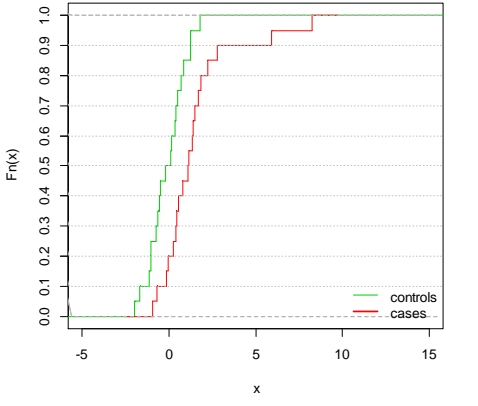
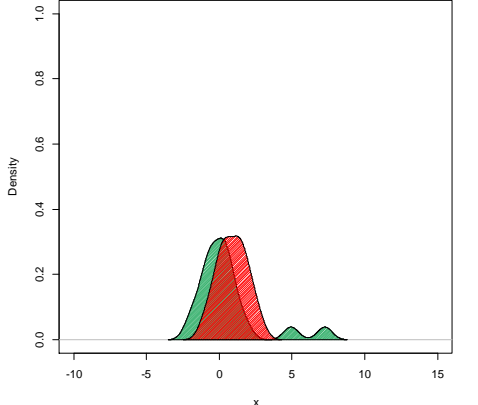
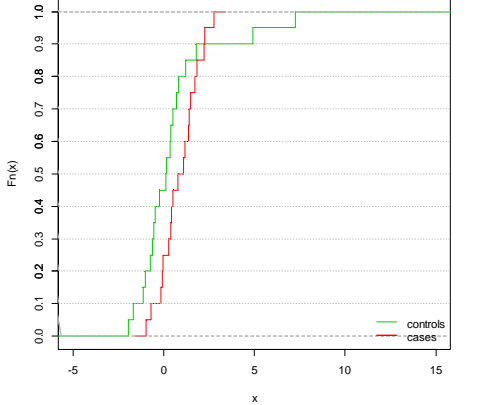
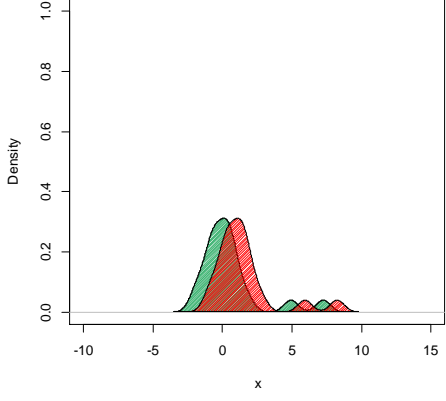
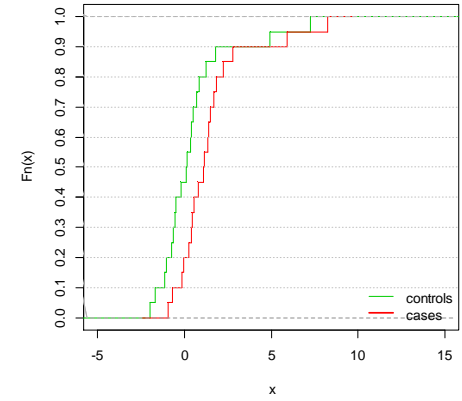
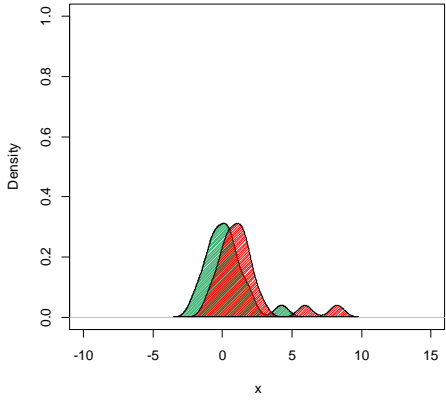
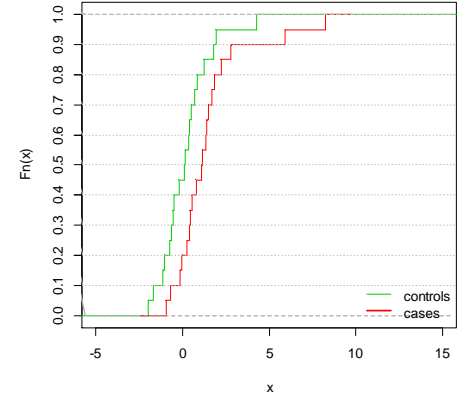
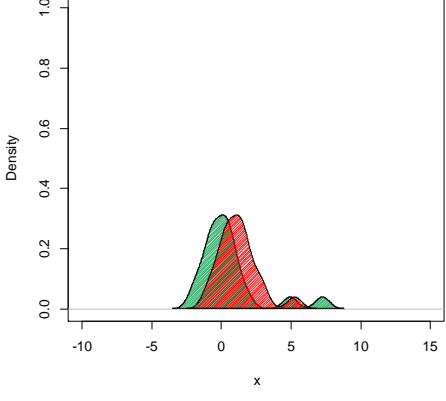
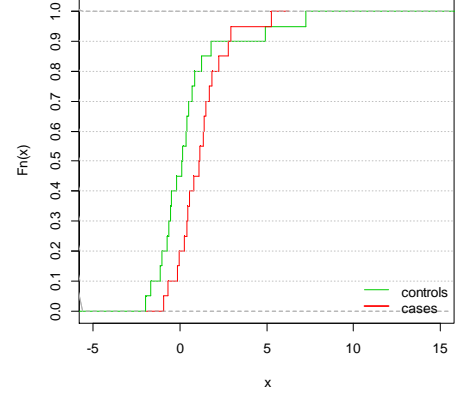
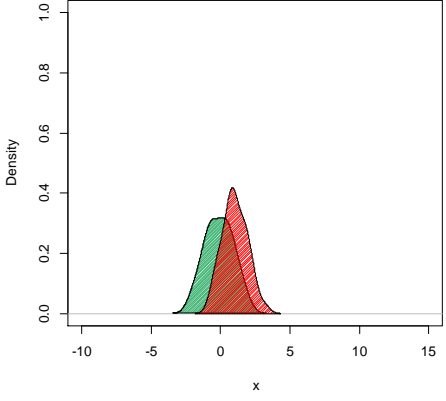
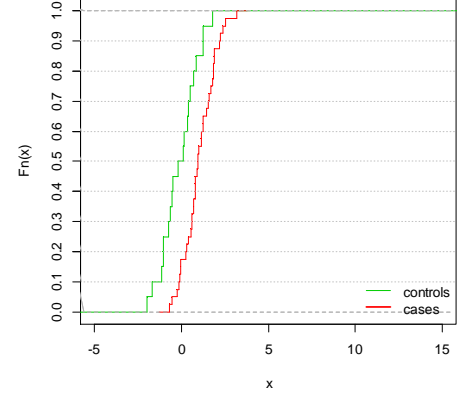
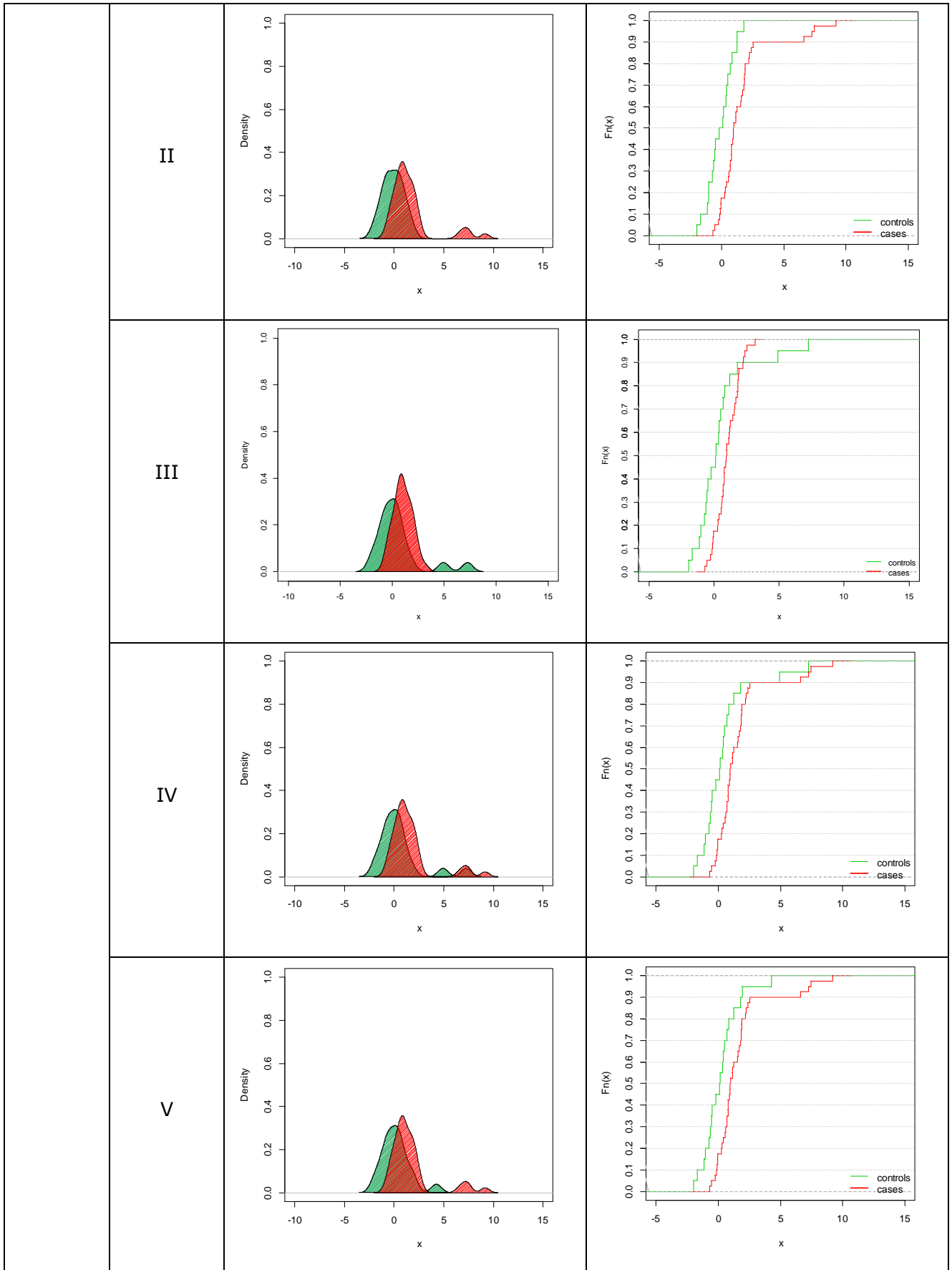


Figure A2.1. Plots of density distributions and ECDFs ($\delta=1$ and $\lambda=0.80$) of the two simulated samples (cases in red and controls in green) for the six selected patterns and three sample size settings ($m=20$ vs $n=20$, $m=20$ vs $n=40$ and $m=40$ vs $n=20$): I. two normals; II. normal vs mixture ($sh=6$); III. mixture ($sh=6$) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2=6$); V. two mixtures with different shifts ($sh_1=3 < sh_2=6$); VI. two mixtures with different shifts ($sh_1=6 > sh_2=3$).

A2.2 Density distributions and ECDFs for $\lambda=0.95$

(m,n)	Pattern	Density distributions	ECDFs
(20,20)	I		
	II		
	III		

	IV		
	V		
	VI		
(20,40)	I		



	VI		
(40,20)	I		
	II		
	III		

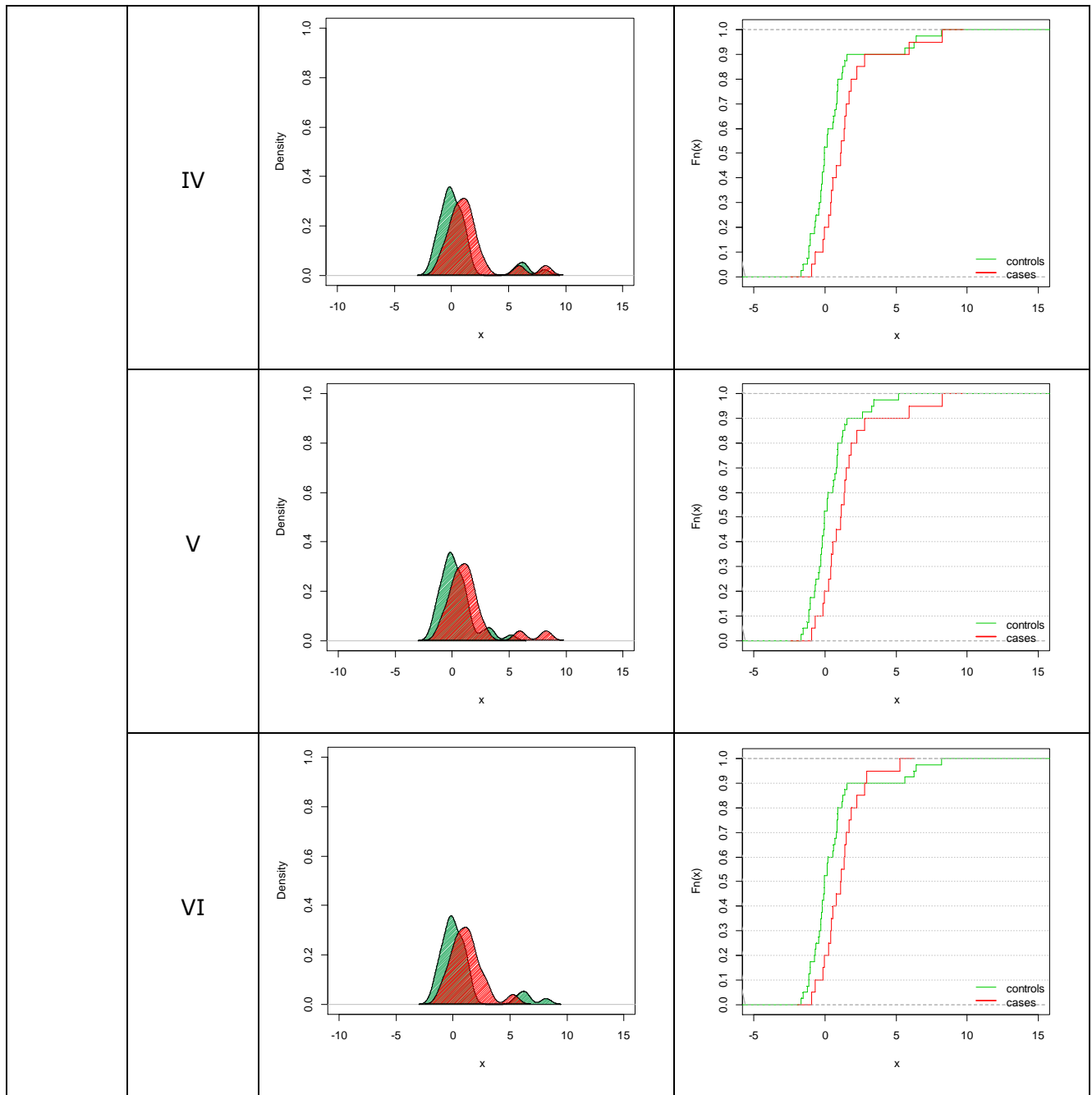
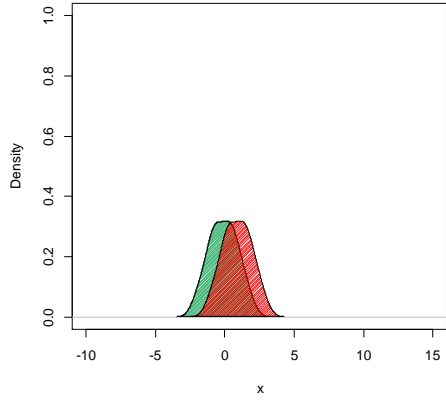
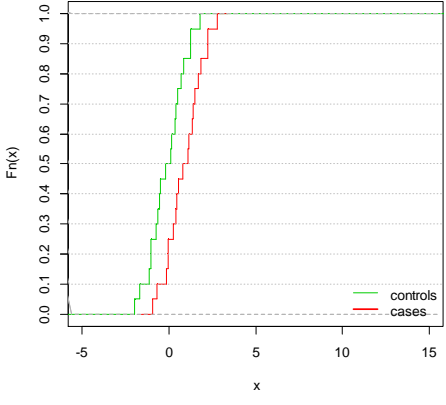
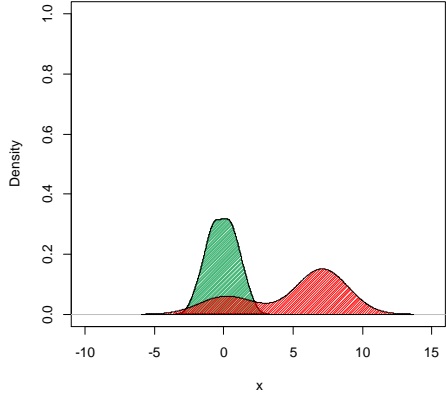
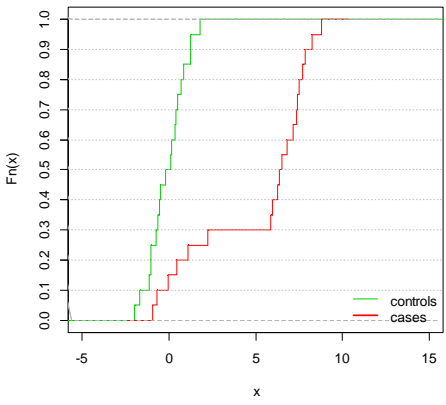
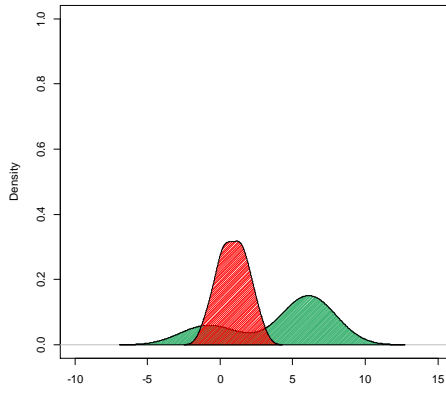
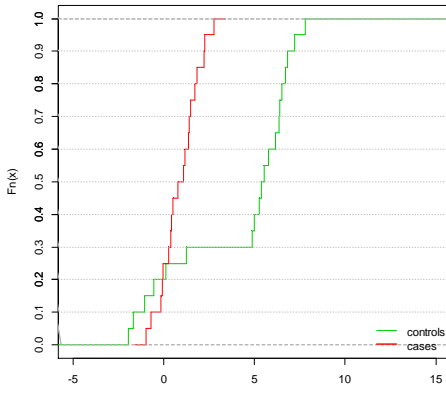
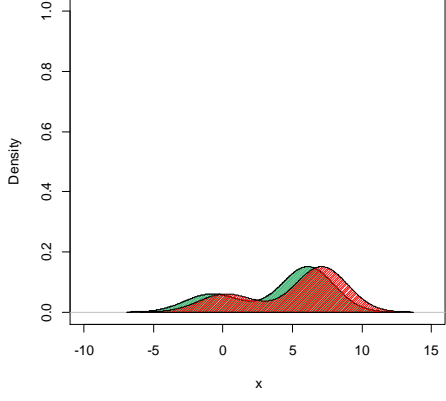
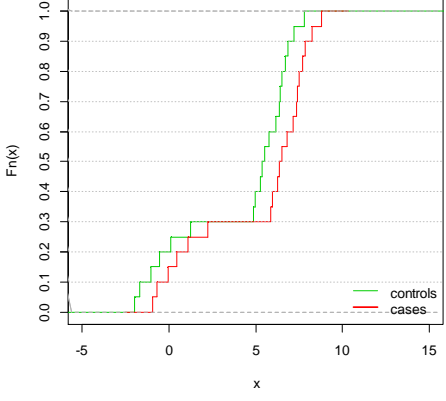
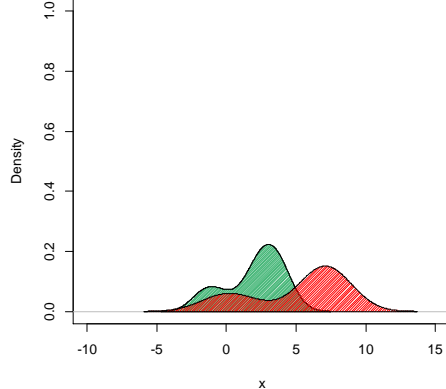
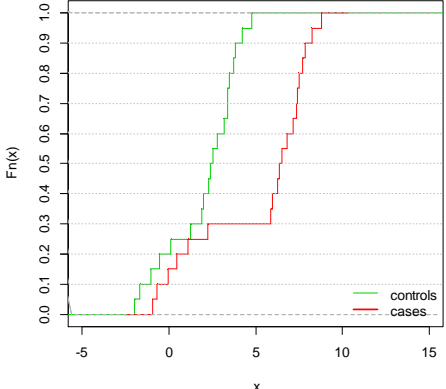
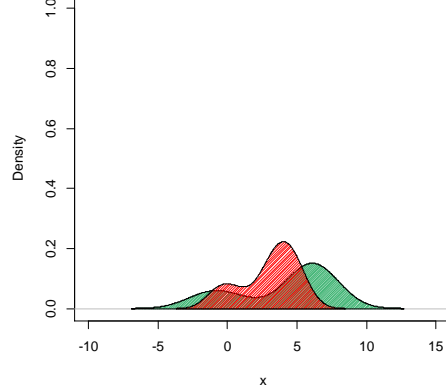
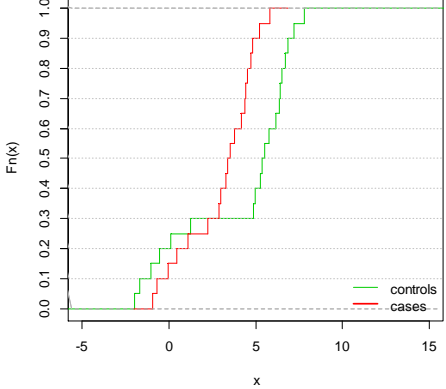
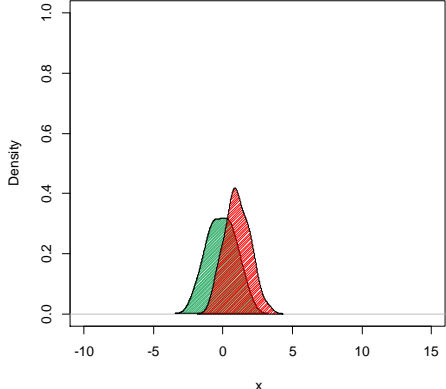
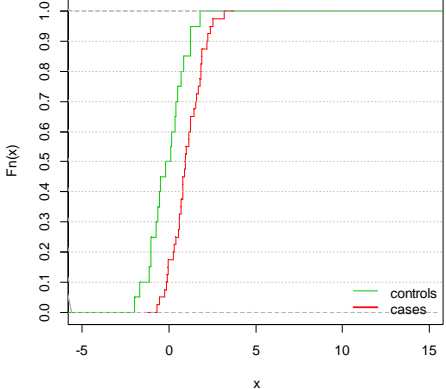
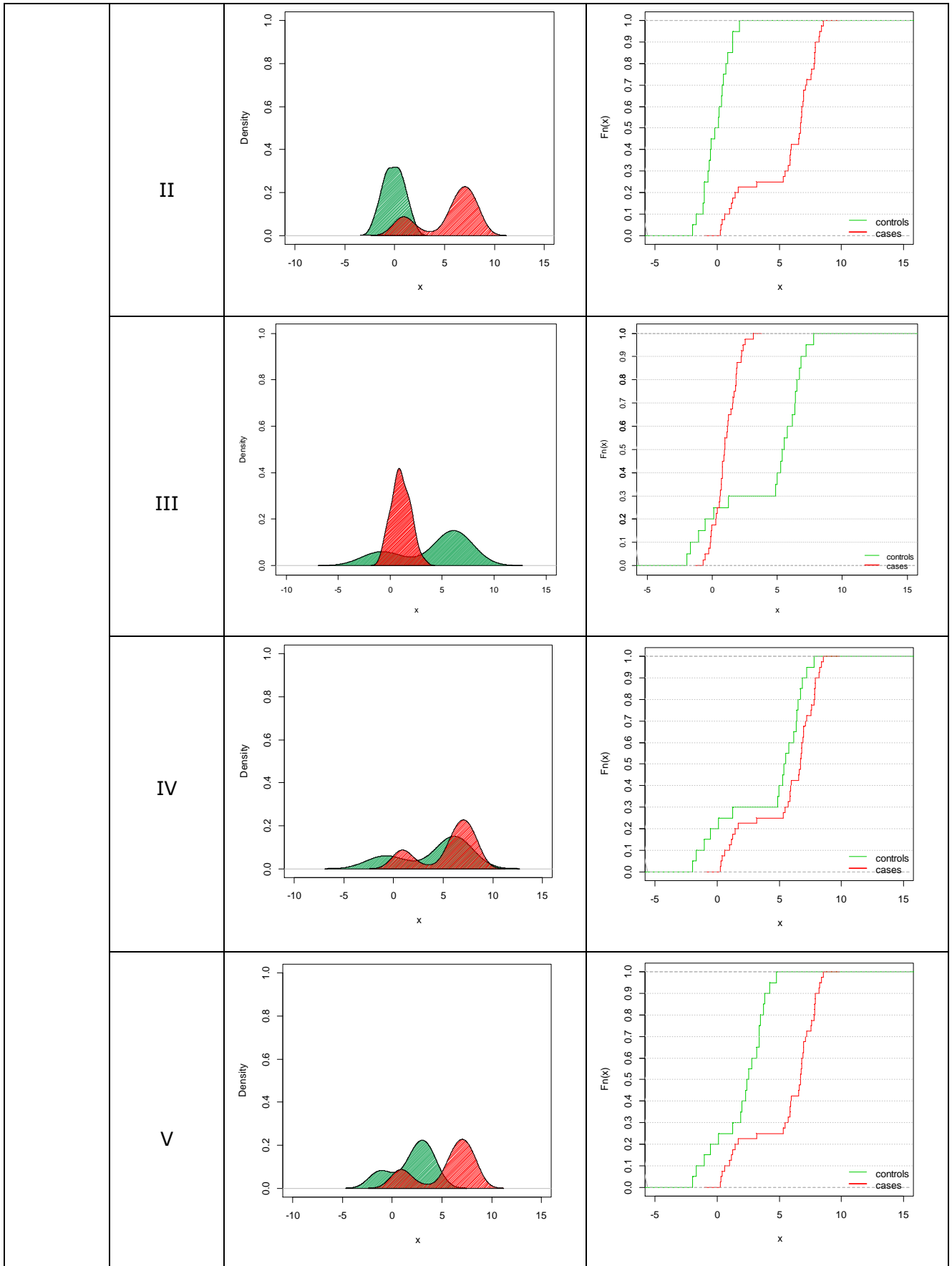


Figure A2.2. Plots of density distributions and ECDFs ($\delta=1$ and $\lambda=0.95$) of the two simulated samples (cases in red and controls in green) for the six selected patterns and three sample size settings ($m=20$ vs $n=20$, $m=20$ vs $n=40$ and $m=40$ vs $n=20$): I. two normals; II. normal vs mixture ($sh=6$); III. mixture ($sh=6$) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2=6$); V. two mixtures with different shifts ($sh_1=3 < sh_2=6$); VI. two mixtures with different shifts ($sh_1=6 > sh_2=3$).

A2.3 Density distributions and ECDFs for $\lambda=0.20$

(m,n)	Pattern	Density distributions	ECDFs
(20,20)	I		
	II		
	III		

	IV		
	V		
	VI		
(20,40)	I		



	VI		
(40,20)	I		
	II		
	III		

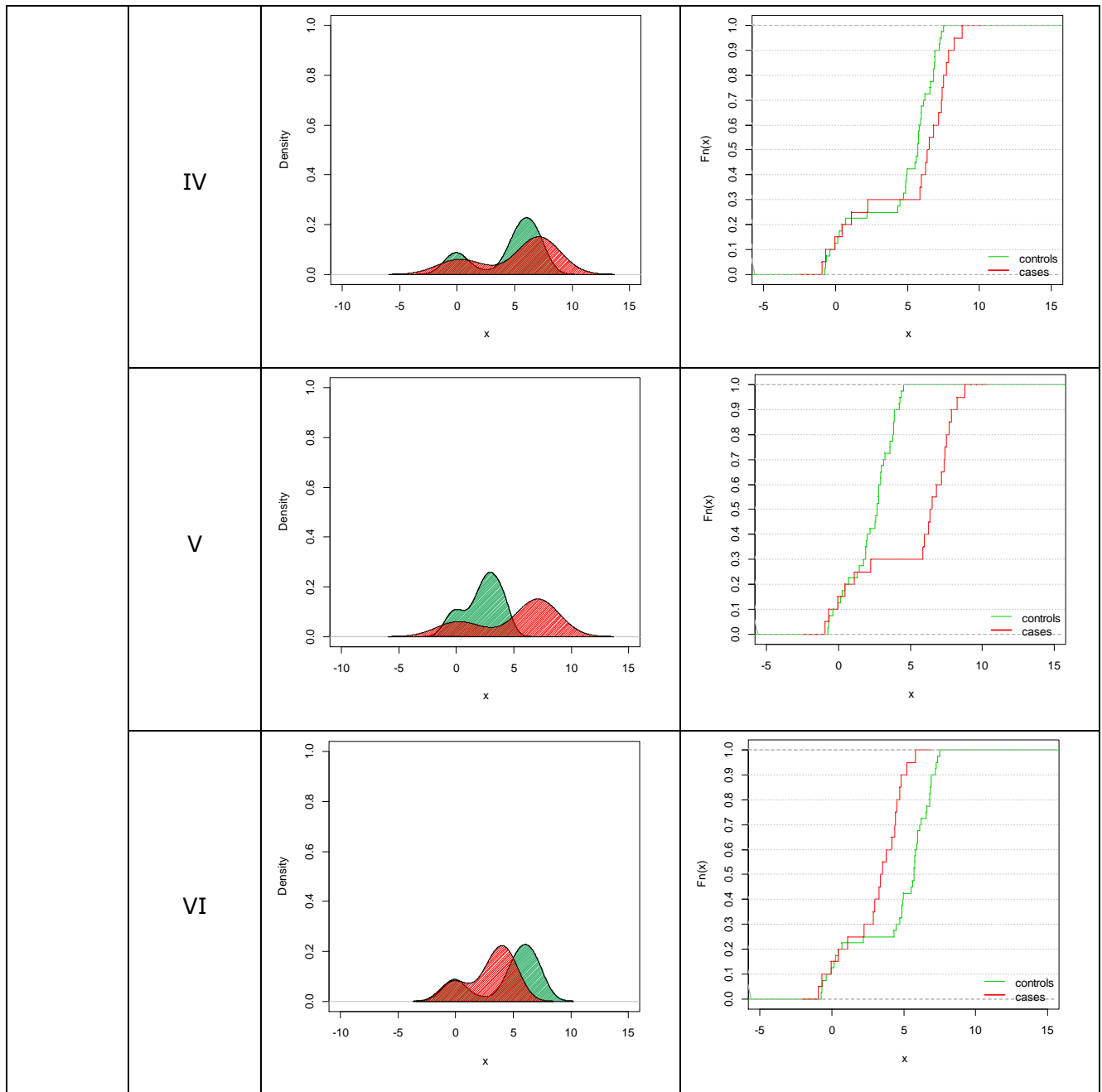
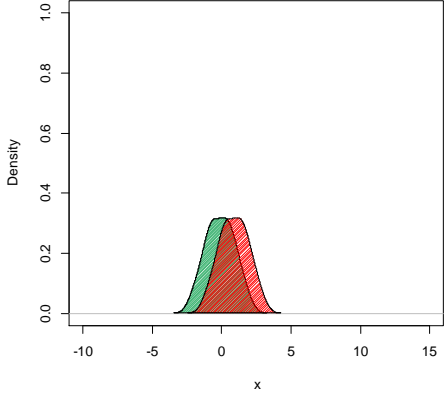
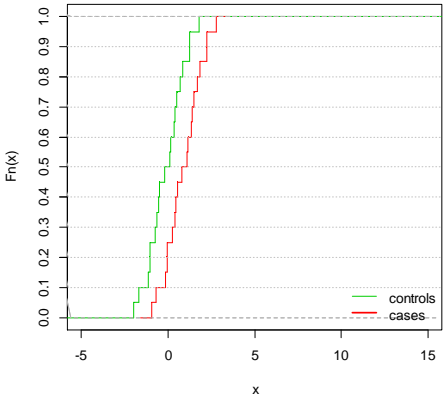
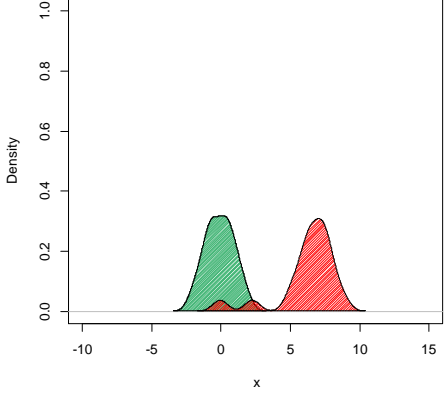
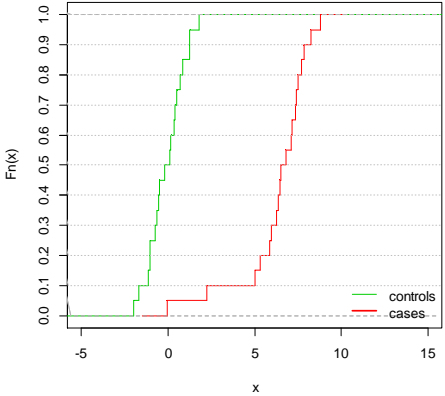
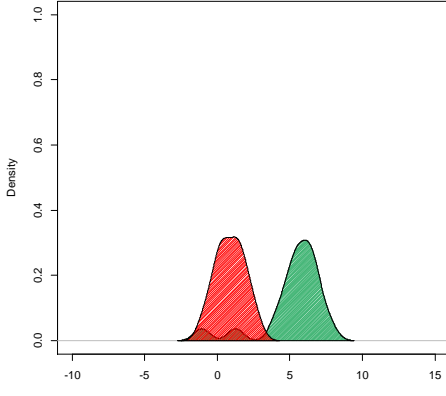
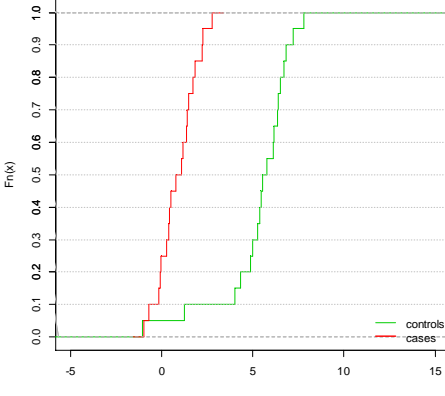
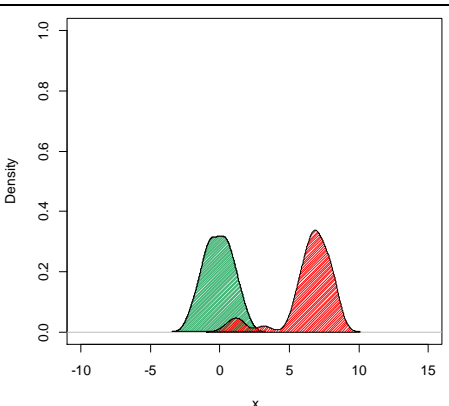
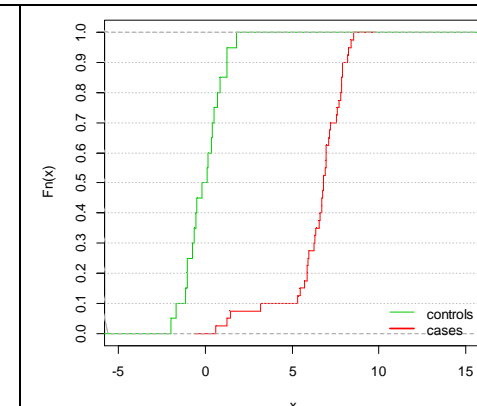
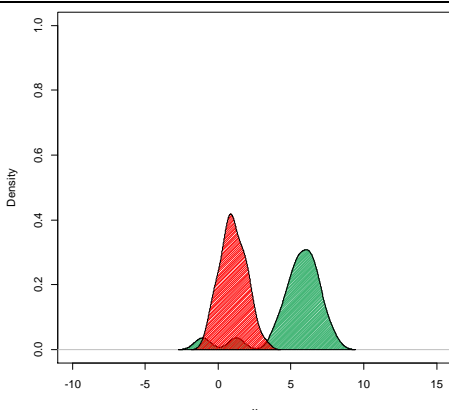
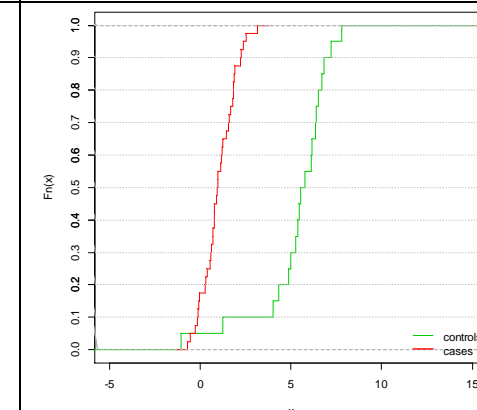
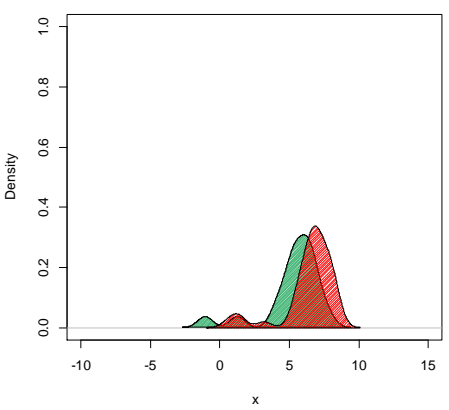
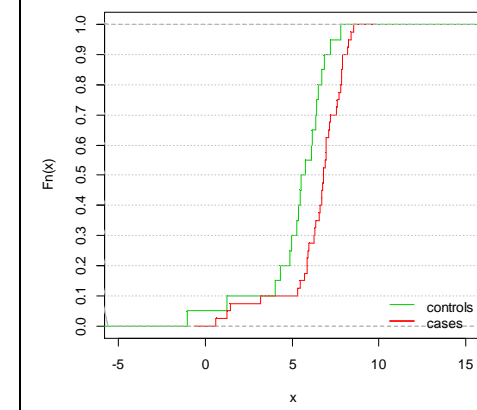
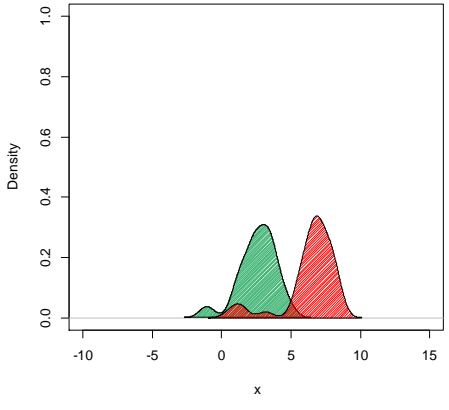
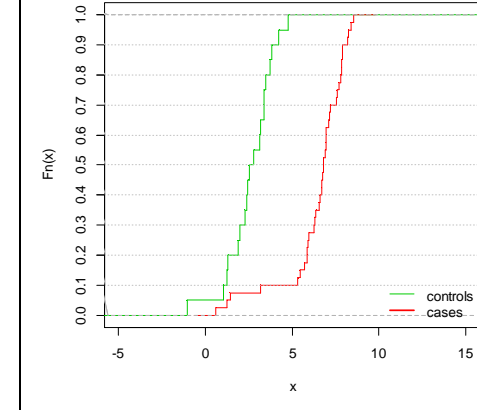


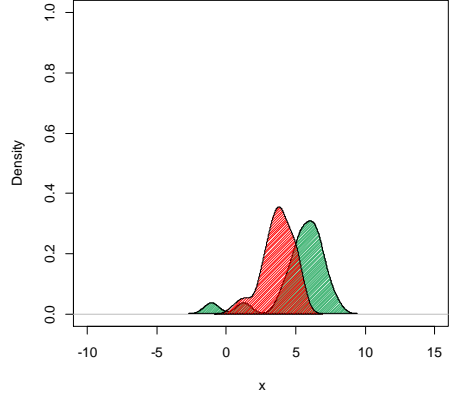
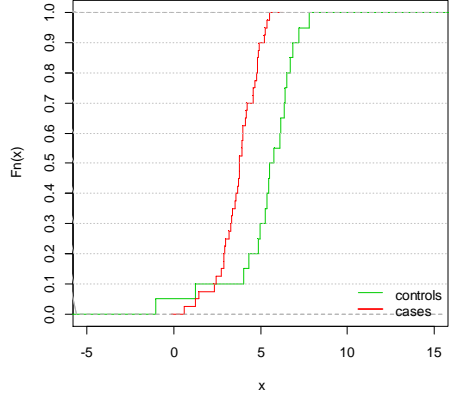
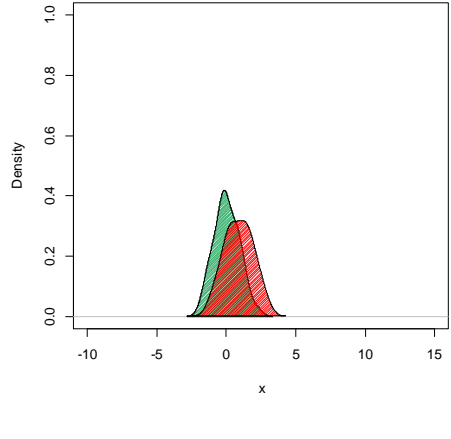
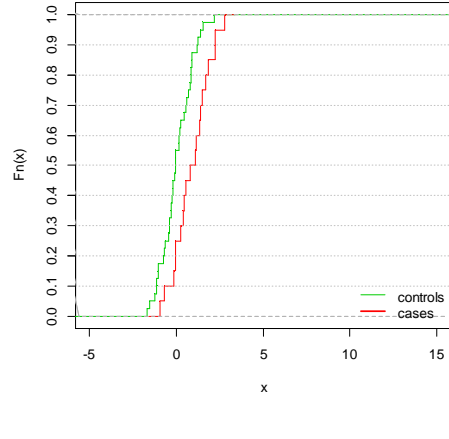
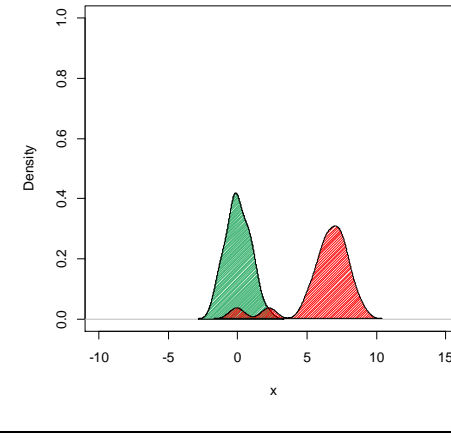
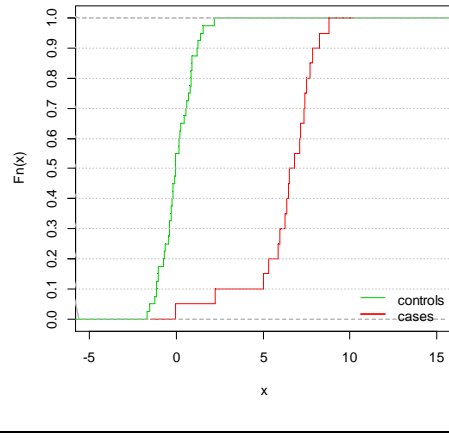
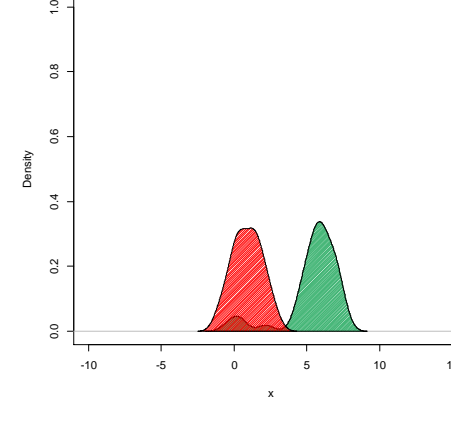
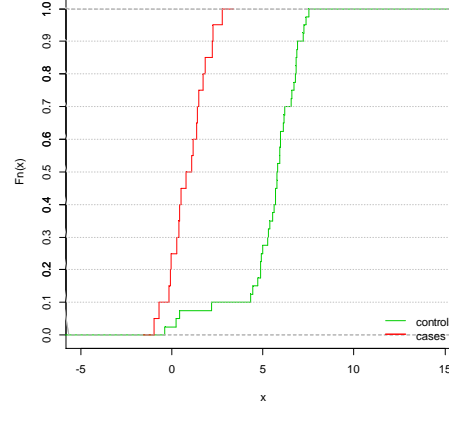
Figure A2.3. Plots of density distributions and ECDFs ($\delta=1$ and $\lambda=0.20$) of the two simulated samples (cases in red and controls in green) for the six selected patterns and three sample size settings ($m=20$ vs $n=20$, $m=20$ vs $n=40$ and $m=40$ vs $n=20$): I. two normals; II. normal vs mixture ($sh=6$); III. mixture ($sh=6$) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2=6$); V. two mixtures with different shifts ($sh_1=3 < sh_2=6$); VI. two mixtures with different shifts ($sh_1=6 > sh_2=3$).

A2.4 Density distributions and ECDFs for $\lambda=0.05$

(m,n)	Pattern	Density distributions	ECDFs
(20,20)	I		
	II		
	III		

	IV		
	V		
	VI		
(20,40)	I		

	II		
	III		
	IV		
	V		

	VI		
(40,20)	I		
	II		
	III		

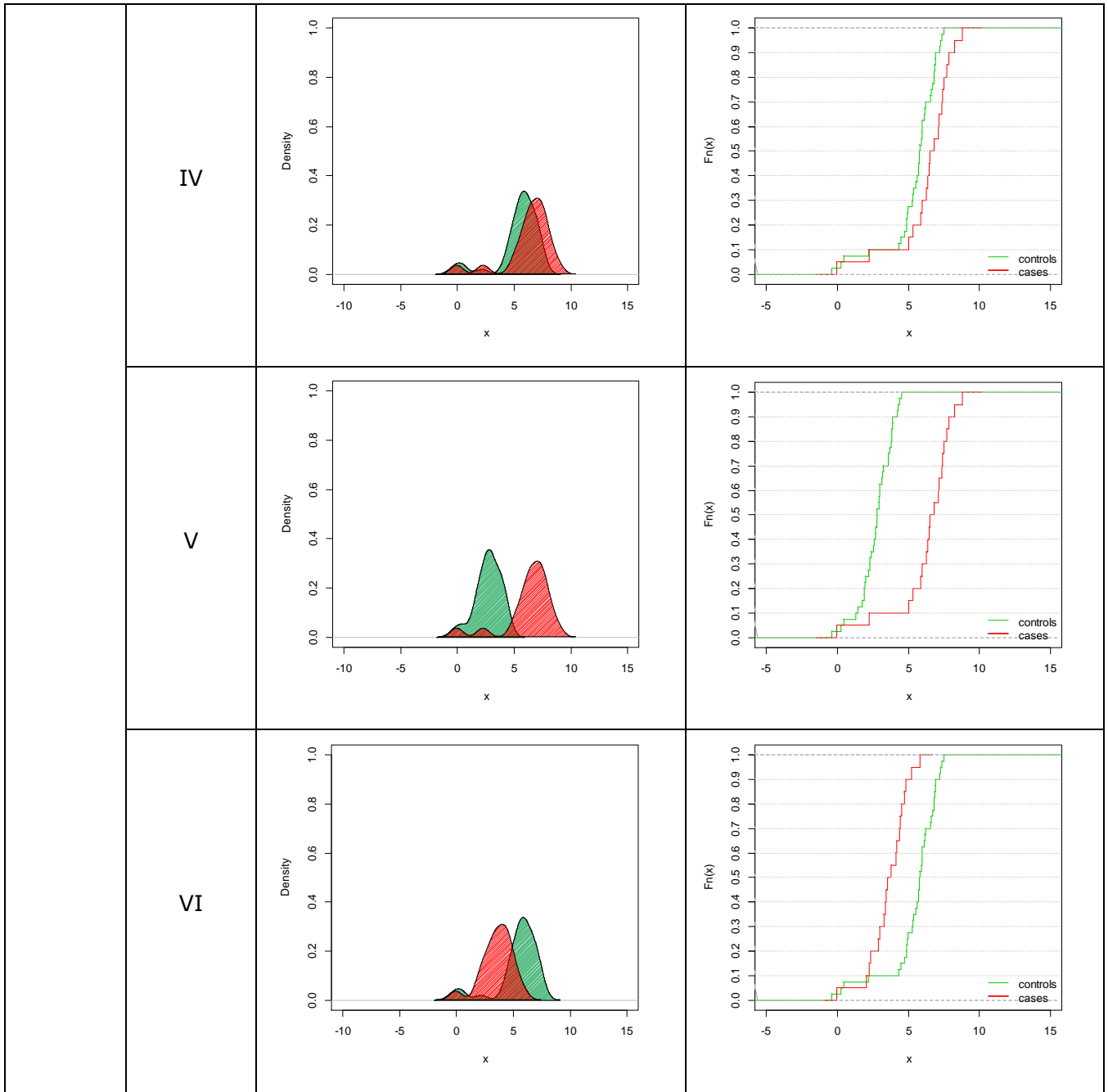


Figure A2.4. Plots of density distributions and ECDFs ($\delta=1$ and $\lambda=0.05$) of the two simulated samples (cases in red and controls in green) for the six selected patterns and three sample size settings ($m=20$ vs $n=20$, $m=20$ vs $n=40$ and $m=40$ vs $n=20$): I. two normals; II. normal vs mixture ($sh=6$); III. mixture ($sh=6$) vs normal; IV. two mixtures with equal shifts ($sh_1=sh_2=6$); V. two mixtures with different shifts ($sh_1=3 < sh_2=6$); VI. two mixtures with different shifts ($sh_1=6 > sh_2=3$).

Appendix 3. Simulation results in terms of size and power ($\alpha = 0.05$)

Tables of the simulation results for the selected patterns under H_0 ($\delta = 0$) and H_1 ($\delta = 1$), divided primarily according to the mixture weight λ ($\lambda = 0.80$, $\lambda = 0.95$, $\lambda = 0.20$ and $\lambda = 0.05$) and, secondly, according to the sample size (m,n) ($(m,n) = (20,20)$, $(m,n) = (20,40)$, $(m,n) = (40,20)$). Characteristics of the two distributions X and Y and size and power estimates are reported.

A3.1 Size and power for $\lambda=0.80$

H_0 patterns ($\delta=0$)

Table A3.1.1. Characteristics of X and Y distributions

Pattern	m_{11}	m_{12}	m_{21}	m_{22}	M_1	M_2	MED_1	MED_2	σ_1	σ_2
I	0	0	0	0	0.0	0.0	0	0	1	1
IV	0	3	0	3	0.6	0.6	0.3	0.3	1.6	1.6
IV	0	6	0	6	1.2	1.2	0.3	0.3	2.6	2.6

Table A3.1.2. Size estimates with $(m,n) = (20,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.053	0.052	0.055	0.051	0.055	0.037	0.055	0.056	0.059	0.054	0.054
IV	0.050	0.049	0.051	0.054	0.049	0.054	0.034	0.053	0.053	0.059	0.052	0.052
IV	0.049	0.047	0.050	0.053	0.049	0.053	0.035	0.053	0.054	0.060	0.051	0.051

Table A3.1.3. Size estimates with $(m,n) = (20,40)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.054	0.050	0.048	0.045	0.049	0.041	0.051	0.051	0.052	0.052	0.051
IV	0.053	0.051	0.049	0.052	0.049	0.052	0.042	0.051	0.051	0.052	0.053	0.053
IV	0.049	0.053	0.048	0.051	0.049	0.052	0.042	0.052	0.052	0.052	0.053	0.051

Table A3.1.4. Size estimates with $(m,n) = (40,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.049	0.048	0.046	0.047	0.044	0.047	0.044	0.050	0.048	0.051	0.048	0.048
IV	0.052	0.053	0.051	0.053	0.050	0.053	0.042	0.051	0.052	0.055	0.051	0.052
IV	0.052	0.054	0.050	0.052	0.049	0.053	0.042	0.050	0.052	0.055	0.051	0.052

H₁ patterns ($\delta=1$)

Table A3.1.5. Characteristics of X and Y distributions

Pattern	m ₁₁	m ₁₂	m ₂₁	m ₂₂	M ₁	M ₂	MED ₁	MED ₂	σ_1	σ_2
I	0	0	1	1	0	1	0	1	1	1
II	0	0	1	7	0	2.2	0	1.3	1	2.6
III	0	6	1	1	1.2	1	0.3	1	2.6	1
IV	0	6	1	7	1.2	2.2	0.3	1.3	2.6	2.6
V	0	3	1	7	0.6	2.2	0.3	1.3	1.6	2.6
VI	0	6	1	4	1.2	1.6	0.3	1.3	2.6	1.6

Table A3.1.6. Power estimates with (m,n) = (20,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.869	0.868	0.849	0.757	0.745	0.755	0.698	0.834	0.848	0.752	0.839	0.834
II	0.981	0.980	0.953	0.916	0.909	0.915	0.865	0.945	0.954	0.924	0.962	0.963
III	0.061	0.057	0.223	0.617	0.603	0.614	0.298	0.375	0.435	0.528	0.542	0.559
IV	0.213	0.212	0.482	0.492	0.477	0.491	0.427	0.518	0.530	0.489	0.527	0.525
V	0.639	0.634	0.584	0.493	0.476	0.488	0.442	0.570	0.603	0.567	0.649	0.657
VI	0.117	0.116	0.360	0.671	0.656	0.668	0.432	0.532	0.574	0.557	0.623	0.630

Table A3.1.7. Power estimates with (m,n) = (20,40)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.947	0.944	0.936	0.881	0.875	0.881	0.858	0.925	0.932	0.857	0.921	0.918
II	0.997	0.999	0.992	0.979	0.977	0.979	0.965	0.989	0.992	0.979	0.992	0.992
III	0.145	0.058	0.299	0.796	0.790	0.796	0.461	0.528	0.690	0.715	0.872	0.871
IV	0.284	0.298	0.599	0.680	0.673	0.680	0.598	0.649	0.687	0.600	0.688	0.679
V	0.748	0.830	0.711	0.651	0.643	0.650	0.610	0.700	0.745	0.683	0.788	0.794
VI	0.181	0.133	0.470	0.857	0.852	0.856	0.613	0.693	0.796	0.709	0.889	0.883

Table A3.1.8. Power estimates with (m,n) = (40,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.948	0.942	0.937	0.888	0.882	0.888	0.865	0.927	0.935	0.864	0.927	0.923
II	0.998	0.991	0.989	0.979	0.979	0.980	0.962	0.985	0.991	0.973	0.994	0.994
III	0.016	0.064	0.257	0.686	0.675	0.684	0.432	0.505	0.539	0.657	0.588	0.633
IV	0.279	0.248	0.624	0.610	0.597	0.608	0.617	0.674	0.675	0.637	0.655	0.656
V	0.822	0.724	0.732	0.643	0.631	0.642	0.634	0.726	0.763	0.738	0.829	0.838
VI	0.086	0.123	0.470	0.741	0.734	0.741	0.619	0.695	0.707	0.700	0.694	0.722

A3.2 Size and power for $\lambda=0.95$

H_0 patterns ($\delta=0$)

Table A3.2.1. Characteristics of X and Y distributions

Pattern	m_{11}	m_{12}	m_{21}	m_{22}	M_1	M_2	MED_1	MED_2	σ_1	σ_2
I	0	0	0	0	0	0	0	0	1	1
IV	0	3	0	3	0.1	0.1	0.1	0.1	1.2	1.2
IV	0	6	0	6	0.3	0.3	0.1	0.1	1.6	1.6

Table A3.2.2. Size estimates with $(m,n) = (20,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.053	0.052	0.055	0.051	0.055	0.037	0.055	0.056	0.059	0.054	0.054
IV	0.053	0.052	0.052	0.056	0.051	0.055	0.036	0.054	0.054	0.063	0.055	0.055
IV	0.047	0.044	0.052	0.056	0.051	0.056	0.036	0.054	0.055	0.063	0.055	0.055

Table A3.2.3. Size estimates with $(m,n) = (20,40)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.054	0.050	0.048	0.045	0.049	0.041	0.051	0.051	0.052	0.052	0.051
IV	0.053	0.053	0.052	0.050	0.047	0.050	0.044	0.054	0.053	0.053	0.053	0.053
IV	0.049	0.054	0.051	0.049	0.046	0.049	0.044	0.054	0.053	0.053	0.053	0.053

Table A3.2.4. Size estimates with $(m,n) = (40,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.049	0.048	0.046	0.047	0.044	0.047	0.044	0.050	0.048	0.051	0.048	0.048
IV	0.048	0.048	0.049	0.048	0.045	0.049	0.045	0.051	0.051	0.050	0.048	0.047
IV	0.046	0.050	0.049	0.048	0.046	0.049	0.045	0.051	0.051	0.050	0.048	0.047

H₁ patterns ($\delta=1$)

Table A3.2.5. Characteristics of X and Y distributions

Pattern	m ₁₁	m ₁₂	m ₂₁	m ₂₂	M ₁	M ₂	MED ₁	MED ₂	σ_1	σ_2
I	0	0	1	1	0	1	0	1	1	1
II	0	0	1	7	0	1.3	0	1.1	1	1.6
III	0	6	1	1	0.3	1	0.1	1	1.6	1
IV	0	6	1	7	0.3	1.3	0.1	1.1	1.6	1.6
V	0	3	1	7	0.1	1.3	0.1	1.1	1.2	1.6
VI	0	6	1	4	0.3	1.1	0.1	1.1	1.6	1.2

Table A3.2.6. Power estimates with (m,n) = (20,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.869	0.868	0.849	0.757	0.745	0.755	0.698	0.834	0.848	0.752	0.839	0.834
II	0.904	0.901	0.883	0.804	0.794	0.803	0.745	0.869	0.880	0.799	0.879	0.877
III	0.419	0.417	0.697	0.638	0.623	0.636	0.589	0.720	0.723	0.619	0.690	0.680
IV	0.499	0.497	0.750	0.677	0.661	0.674	0.634	0.761	0.763	0.663	0.729	0.717
V	0.729	0.726	0.770	0.682	0.666	0.680	0.636	0.768	0.772	0.669	0.745	0.737
VI	0.485	0.483	0.743	0.680	0.663	0.678	0.634	0.761	0.764	0.663	0.733	0.722

Table A3.2.7. Power estimates with (m,n) = (20,40)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.947	0.944	0.936	0.881	0.875	0.881	0.858	0.925	0.932	0.857	0.921	0.918
II	0.961	0.978	0.957	0.915	0.910	0.914	0.892	0.949	0.955	0.898	0.950	0.948
III	0.547	0.460	0.811	0.803	0.796	0.804	0.765	0.833	0.849	0.747	0.856	0.846
IV	0.614	0.596	0.858	0.829	0.822	0.828	0.804	0.866	0.876	0.788	0.853	0.840
V	0.829	0.855	0.876	0.827	0.821	0.825	0.805	0.871	0.881	0.793	0.858	0.848
VI	0.615	0.542	0.851	0.834	0.828	0.834	0.803	0.868	0.882	0.789	0.879	0.869

Table A3.2.8. Power estimates with (m,n) = (40,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.948	0.942	0.937	0.888	0.882	0.888	0.865	0.927	0.935	0.864	0.927	0.923
II	0.976	0.955	0.958	0.921	0.918	0.921	0.895	0.949	0.958	0.899	0.958	0.956
III	0.434	0.529	0.834	0.754	0.746	0.755	0.774	0.850	0.841	0.759	0.773	0.773
IV	0.596	0.613	0.879	0.802	0.795	0.802	0.816	0.882	0.878	0.798	0.829	0.824
V	0.874	0.845	0.892	0.816	0.809	0.815	0.817	0.886	0.886	0.802	0.856	0.851
VI	0.539	0.613	0.874	0.798	0.790	0.799	0.816	0.881	0.878	0.798	0.821	0.818

A3.3 Size and power for $\lambda=0.20$

H_0 patterns ($\delta=0$)

Table A3.3.1. Characteristics of X and Y distributions

Pattern	m_{11}	m_{12}	m_{21}	m_{22}	M_1	M_2	MED_1	MED_2	σ_1	σ_2
I	0	0	0	0	0	0	0	0	1	1
IV	0	3	0	3	2.4	2.4	2.7	2.7	1.6	1.6
IV	0	6	0	6	4.8	4.8	5.7	5.7	2.6	2.6

Table A3.3.2. Size estimates with $(m,n) = (20,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.053	0.052	0.055	0.051	0.055	0.037	0.055	0.056	0.060	0.054	0.054
IV	0.049	0.048	0.051	0.051	0.046	0.050	0.035	0.054	0.053	0.058	0.051	0.049
IV	0.046	0.045	0.051	0.052	0.047	0.050	0.036	0.054	0.053	0.058	0.052	0.051

Table A3.3.3. Size estimates with $(m,n) = (20,40)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.054	0.050	0.048	0.045	0.049	0.041	0.051	0.051	0.052	0.052	0.051
IV	0.046	0.046	0.047	0.049	0.047	0.050	0.039	0.049	0.048	0.047	0.045	0.045
IV	0.045	0.049	0.046	0.047	0.045	0.048	0.040	0.048	0.046	0.046	0.044	0.045

Table A3.3.4. Size estimates with $(m,n) = (40,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.049	0.048	0.046	0.047	0.044	0.047	0.044	0.050	0.048	0.051	0.048	0.048
IV	0.049	0.050	0.046	0.048	0.046	0.050	0.042	0.048	0.049	0.052	0.047	0.047
IV	0.050	0.056	0.047	0.049	0.047	0.051	0.043	0.049	0.048	0.052	0.048	0.048

H₁ patterns ($\delta=1$)

Table A3.3.5. Characteristics of X and Y distributions

Pattern	m ₁₁	m ₁₂	m ₂₁	m ₂₂	M ₁	M ₂	MED ₁	MED ₂	σ_1	σ_2
I	0	0	1	1	0.0	1.0	0	1	1	1
II	0	0	1	7	0.0	5.8	0	6.7	1	2.6
III	0	6	1	1	4.8	1.0	5.7	1	2.6	1
IV	0	6	1	7	4.8	5.8	5.7	6.7	2.6	2.6
V	0	3	1	7	2.4	5.8	2.7	6.7	1.6	2.6
VI	0	6	1	4	4.8	3.4	5.7	3.7	2.6	1.6

Table A3.3.6. Power estimates with (m,n) = (20,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.869	0.868	0.849	0.757	0.745	0.755	0.698	0.834	0.848	0.752	0.839	0.834
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	0.979	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.227	0.226	0.500	0.499	0.482	0.494	0.437	0.537	0.547	0.495	0.537	0.533
V	0.995	0.995	0.979	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	1.000
VI	0.516	0.512	0.818	0.996	0.995	0.996	0.950	0.976	0.985	0.976	0.988	0.988

Table A3.3.7. Power estimates with (m,n) = (20,40)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.947	0.944	0.936	0.881	0.875	0.881	0.858	0.925	0.932	0.857	0.921	0.918
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	0.991	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.284	0.253	0.627	0.609	0.598	0.609	0.619	0.677	0.675	0.638	0.656	0.656
V	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
VI	0.682	0.555	0.880	1.000	1.000	1.000	0.989	0.996	0.999	0.994	0.999	0.999

Table A3.3.8. Power estimates with (m,n) = (40,20)

Pattern	T	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.948	0.942	0.937	0.888	0.882	0.888	0.865	0.927	0.935	0.864	0.927	0.923
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.288	0.299	0.614	0.686	0.677	0.686	0.608	0.654	0.692	0.601	0.696	0.689
V	0.999	0.996	0.990	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
VI	0.589	0.717	0.943	0.999	0.999	0.999	0.995	0.998	0.998	0.997	0.997	0.998

A3.4 Size and power for $\lambda=0.05$

H_0 patterns ($\delta=0$)

Table A3.4.1. Characteristics of X and Y distributions

Pattern	m_{11}	m_{12}	m_{21}	m_{22}	M_1	M_2	MED_1	MED_2	σ_1	σ_2
I	0	0	0	0	0	0	0	0	1	1
IV	0	3	0	3	2.8	2.8	2.9	2.9	1.2	1.2
IV	0	6	0	6	5.7	5.7	5.9	5.9	1.6	1.6

Table A3.4.2. Size estimates with $(m,n) = (20,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.053	0.052	0.055	0.051	0.055	0.037	0.055	0.056	0.059	0.054	0.054
IV	0.049	0.049	0.052	0.054	0.050	0.055	0.038	0.056	0.056	0.058	0.054	0.053
IV	0.046	0.044	0.052	0.054	0.050	0.054	0.038	0.056	0.055	0.058	0.053	0.052

Table A3.4.3. Size estimates with $(m,n) = (20,40)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.053	0.054	0.050	0.048	0.045	0.049	0.041	0.051	0.051	0.052	0.052	0.051
IV	0.047	0.047	0.047	0.049	0.046	0.049	0.041	0.051	0.050	0.049	0.046	0.046
IV	0.042	0.047	0.047	0.049	0.046	0.049	0.041	0.051	0.050	0.049	0.045	0.045

Table A3.4.4. Size estimates with $(m,n) = (40,20)$

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z_K	Z_C	Z_A
I	0.049	0.048	0.046	0.047	0.044	0.047	0.044	0.050	0.048	0.051	0.04	0.048
IV	0.049	0.049	0.046	0.051	0.048	0.051	0.043	0.049	0.050	0.049	0.049	0.049
IV	0.047	0.050	0.047	0.050	0.048	0.050	0.043	0.050	0.050	0.050	0.049	0.048

H₁ patterns ($\delta=1$)

Table A3.4.5. Characteristics of X and Y distributions

Pattern	m ₁₁	m ₁₂	m ₂₁	m ₂₂	M ₁	M ₂	MED ₁	MED ₂	σ_1	σ_2
I	0	0	1	1	0	1	0	1	1	1
II	0	0	1	7	0	6.7	0	6.9	1	1.6
III	0	6	1	1	5.7	1	5.9	1	1.6	1
IV	0	6	1	7	5.7	6.7	5.9	6.9	1.6	1.6
V	0	3	1	7	2.8	6.7	2.9	6.9	1.2	1.6
VI	0	6	1	4	5.7	3.8	5.9	3.9	1.6	1.2

Table A3.4.6. Power estimates with (m,n) = (20,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.869	0.868	0.849	0.757	0.745	0.755	0.698	0.834	0.848	0.752	0.839	0.834
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.510	0.507	0.754	0.680	0.666	0.678	0.635	0.761	0.763	0.669	0.735	0.721
V	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
VI	0.942	0.939	0.998	0.999	0.998	0.999	0.997	1.000	1.000	0.998	0.999	0.999

Table A3.4.7. Power estimates with (m,n) = (20,40)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.947	0.944	0.936	0.881	0.875	0.881	0.858	0.925	0.932	0.857	0.921	0.918
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.603	0.621	0.878	0.804	0.794	0.804	0.810	0.880	0.877	0.798	0.829	0.825
V	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
VI	0.981	0.954	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table A3.4.8. Power estimates with (m,n) = (40,20)

Pattern	t	Welch	WMW	PG2	C	C.emp	KS	CvM	AD	Z _K	Z _C	Z _A
I	0.948	0.942	0.937	0.888	0.882	0.888	0.865	0.927	0.935	0.864	0.927	0.923
II	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
III	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IV	0.623	0.604	0.867	0.840	0.833	0.840	0.813	0.875	0.882	0.795	0.862	0.848
V	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
VI	0.985	0.995	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Appendix 4. Real data example: exemplificative kernel density distributions of selected ‘gray-zone’ genes in ER positive (in red) and ER negative (in green) subjects

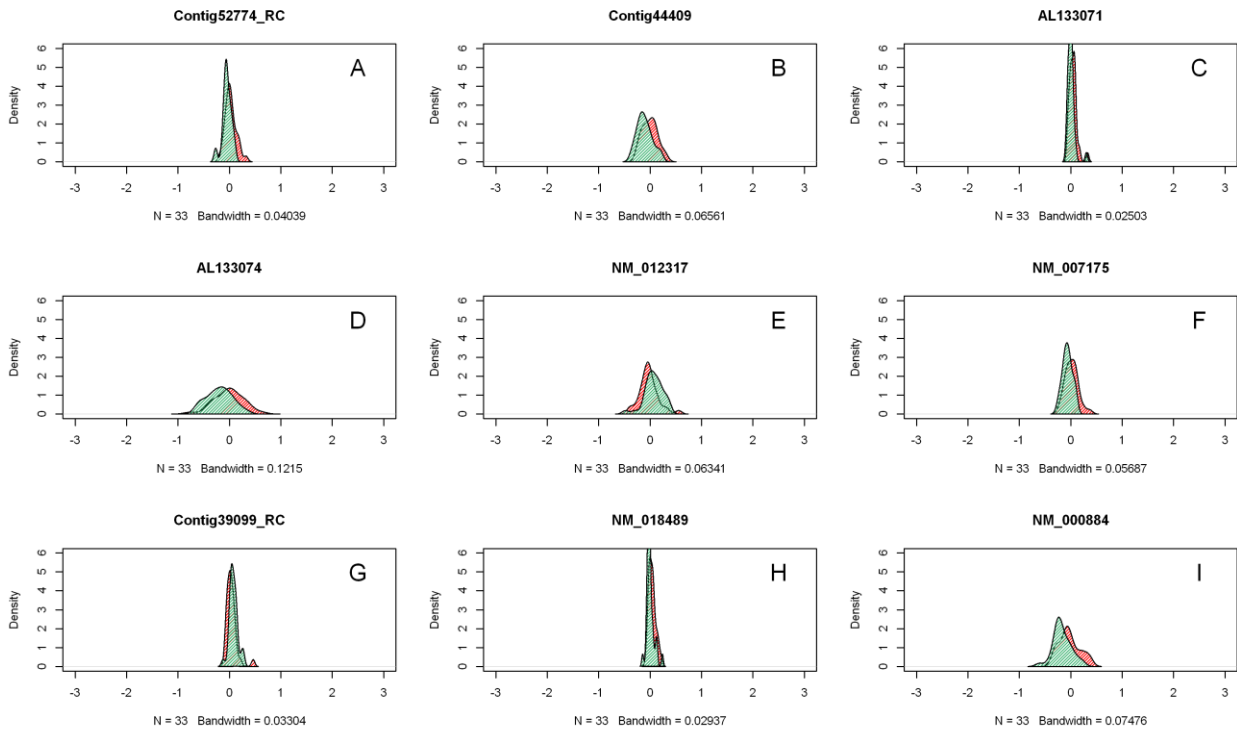


Figure A4.1. Exemplificative density distributions of 9 genes (out of 36 features) separately identified by the WMW test.

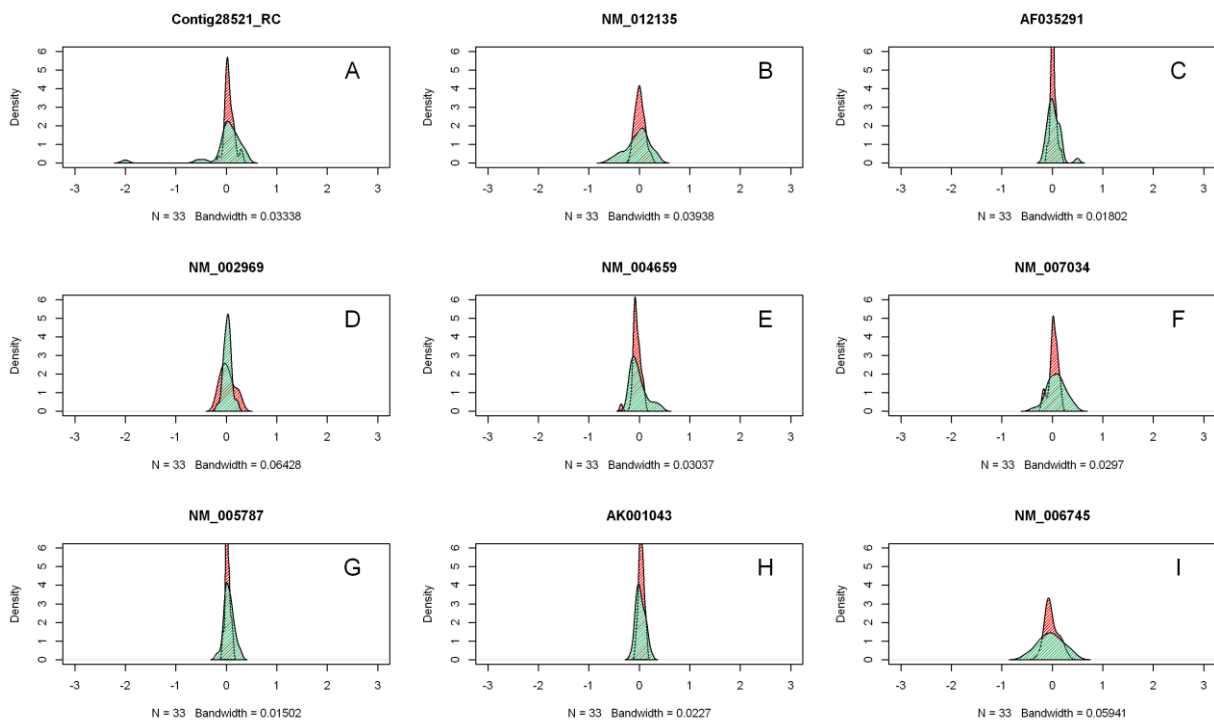


Figure A4.2. Exemplificative density distributions of 9 genes (out of 137 features) separately identified by the PG2 test.

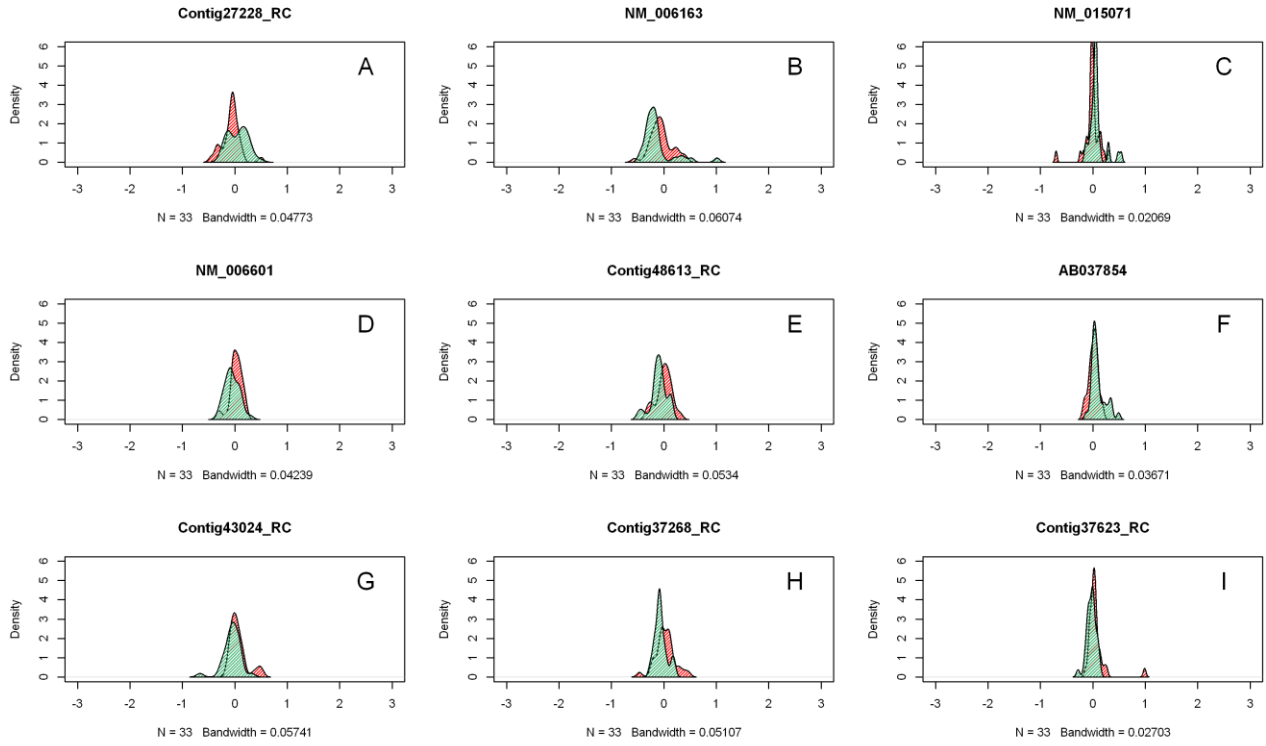


Figure A4.3. Exemplificative density distributions of 9 genes (out of 113 features) separately identified by the AD test.