A vignette for movieROC package:
Visualizing the Decision Rules Underlying Binary Classification

Sonia Pérez-Fernández, Pablo Martínez-Camblor and Norberto Corral-Blanco

Abstract The R package **movieROC** provides visualization tools for understanding the ability of markers to identify a characteristic of interest, complementing the ROC curve representation. This tool accommodates multivariate scenarios and generalizations involving different decision rules. The main contribution of this package is the visualization of the underlying classification regions, with the associated gain in interpretability. Adding the time (videos) as a third dimension, this package facilitates the visualization of binary classification in multivariate problems.

This vignette offers a tutorial introduction to the package. It explains the mathematical formalism underlying the implemented methods and gives the main structure of the `HCC` dataset.

1 Mathematical notation

Continuous markers are surrogate measures for the characteristic under study, or predictors of a potential subsequent event. They are measured in subjects, some of them with the characteristic (*positive*), and some without it (*negative*). A usual practice when the binary classification accuracy of a marker is of interest involves the representation of the *Receiver Operating Characteristic (ROC) curve*, a plot reflecting the trade-off between the sensitivity and the complementary of the specificity.

Mathematically, let ξ and χ be the random variables modeling the marker values in the positive and the negative population, respectively, with $F_{\xi}(\cdot)$ and $F_{\chi}(\cdot)$ their associated cumulative distribution functions (CDFs).

1.1 Regular ROC curve

Assuming that the expected value of the marker is larger in the positive than in the negative population, the standard ROC curve is based on *classification subsets* of the form $s = (c, \infty)$, where c is the so-called *cut-off value or threshold* in the support of the marker X, $\mathcal{S}(X)$. One subject is classified as a positive if its marker value is within this region, and as a negative otherwise. This type of subsets has two important advantages: first, their interpretability is clear; second, for each specificity $1 - t \in [0, 1]$, the corresponding $s_t = (c_t, \infty)$ is univocally defined by $c_t = F_{\chi}^{-1}(1 - t)$.

The *regular, right-sided* or *standard ROC curve* may be seen as the graph depending on cut-off values $c_t \in \mathcal{S}(X) \subseteq \mathbb{R}$, or as a function $\mathcal{R}(\cdot)$ of the complementary of the specificity, $t \in [0, 1]$. That is,

$$\Big\{\big(1 - Sp(c_t), Se(c_t)\big) \text{ with } c_t \in \mathcal{S}(X)\Big\} = \Big\{\big(t, \mathcal{R}(t)\big) \text{ with } t \in [0, 1]\Big\}.$$

In other words, the standard ROC curve associated with the marker X considers classification subsets of the form $s_t = (c_t, \infty)$. As a result, it may be represented by the following equivalences:

$$\begin{aligned}\Big\{\big(\mathbb{P}(\chi \in s_t), \mathbb{P}(\xi \in s_t)\big) : s_t = (c_t, \infty)\ , c_t \in \mathcal{S}(X)\Big\} &= \Big\{\big(1 - F_{\chi}(c_t), 1 - F_{\xi}(c_t)\big) : c_t \in \mathcal{S}(X)\Big\} \\ &= \Big\{\big(t, \mathcal{R}_r(t)\big) \text{ with } t \in [0, 1]\Big\} = \Big\{\big(t, 1 - F_{\xi}\big(1 - F_{\chi}^{-1}(1 - t)\big)\big) : t \in [0, 1]\Big\}.\end{aligned}$$

The first expression, in terms of the eligible classification subsets in the standard scenario, is not commonly considered but reflected here with the aim of focusing on the underlying decision rules. For each fixed specificity $1 - t \in [0, 1]$, also called *true-negative rate*, there exists only one subset $s_t = (c_t, \infty)$ reporting such specificity and thus a particular sensitivity, so-called *true-positive rate*. The underlying subsets are thus univocally determined, resulting in a simple correspondence between each point of the ROC curve $\big(t, \mathcal{R}_r(t)\big)$ and its associated classification region $s_t \in \mathcal{I}_r(t)$, where

$$\mathcal{I}_r(t) = \Big\{s_t = (c_t, \infty) : c_t \in \mathcal{S}(X)\ ,\ \mathbb{P}(\chi \in s_t) = t\Big\}$$

is the *right-sided family of eligible classification subsets*.

If the standard assumption is not fulfilled, but the opposite direction is plausible (i.e. higher values of the marker are associated with a higher probability of being a negative), the ROC curve would be defined by the *left-sided family of eligible classification subsets* (Martínez-Camblor et al., 2017):

$$\mathcal{I}_l(t) = \left\{ s_t = (\infty, c_t] : c_t \in \mathcal{S}(X) \ , \ \mathbb{P}(\chi \in s_t) = t \right\},$$

resulting in $\mathcal{R}_l(t) = F_\xi\left(F_\chi^{-1}(t)\right)$, $t \in [0,1]$. The decision rules are also univocally defined in this case, being $s_t = (-\infty, c_t] = (-\infty, F_\chi^{-1}(t)]$ the classification subset which reports a false-positive rate t.

Information related to the ROC curve is frequently summarized by numerical indices such as the AUC (Hanley and McNeil, 1982) and the *Youden index* (Youden, 1950). The classification rule derived from the Youden index, $s_{t_\mathcal{Y}}$ where $t_\mathcal{Y} = \arg\max_{t\in[0,1]} \left\{\mathcal{R}(t) - t\right\}$, is commonly employed when there is no preference or upper limit for misclassification errors in positive and negative subjects.

Estimation In practice, let $(\xi_1, \xi_2, \ldots, \xi_n)$ and $(\chi_1, \chi_2, \ldots, \chi_m)$ be two independent and identically distributed (i.i.d.) samples from the positive and the negative population, respectively.

The most direct estimator for the ROC curve is the *empirical estimator* (Hsieh and Turnbull, 1996), which directly substitutes the empirical CDFs, $\hat{F}_\xi(\cdot)$ and $\hat{F}_\chi(\cdot)$, for their theoretical counterparts, $F_\xi(\cdot)$ and $F_\chi(\cdot)$, resulting in:

$$\widehat{\mathcal{R}}_r(t) = 1 - \hat{F}_\xi(1 - \hat{F}_\chi^{-1}(1-t)),$$

where $\hat{F}_\xi(x) = \dfrac{\#\left(\xi_i \leq x\right)}{n}$ and $\hat{F}_\chi^{-1}(p) = \min\left\{x \in \mathcal{S}(X) \text{ such that } \hat{F}_\chi(x) = \dfrac{\#\left(\chi_j \leq x\right)}{m} \geq p\right\}$.

This estimation method is implemented in the **movieROC** package, accompanied by its corresponding summary indices: the empirical AUC and the empirical Youden index (see code in Section 3.1 and Figure 3 [right, black line]).

Alternatively, semiparametric approaches based on kernel density estimation for the involved distributions may be considered (Zou et al., 1997). The `plot_densityROC()` function provides plots for both right- and left-sided ROC curved estimated by this method (see Figures 3 [left and middle] and Figure 4).

Binormal scenario Besides the empirical estimation addressed above, a special parametric estimate should be mentioned: the binormal one. This assumes that the marker follows a gaussian distribution in both populations; mathematically, $\xi \sim \mathcal{N}(\mu_\xi, \sigma_\xi)$ and $\chi \sim \mathcal{N}(\mu_\chi, \sigma_\chi)$, where μ and σ are the mean and standard deviation, respectively, usually unknown. Therefore, the *binormal right-sided ROC curve* is

$$\mathcal{R}_r^\mathcal{N}(t) = 1 - F_\xi\left(1 - F_\chi^{-1}(1-t)\right) = \Phi\left(a + b\,\Phi^{-1}(t)\right) \ , \ t \in [0,1],$$

where $a = \left(\mu_\xi - \mu_\chi\right)/\sigma_\xi$ and $b = \sigma_\chi/\sigma_\xi$ measure the difference in means and variances, respectively, $\Phi(\cdot)$ is the cumulative distribution function of a standard normal, and $\Phi^{-1}(\cdot)$ its quantile function. By definition, binormal right-sided ROC curve crosses the diagonal if and only if variances differ ($b \neq 1$). Parametric approaches propose plug-in estimators by estimating the unknown parameters while using the known distributions.

1.2 Generalizations for univariate markers

When differences in marker distribution between the two populations are only in location but not in shape, then the classification is direct by using these decision rules. However, when this is not the case, the standard ROC curve may cross the main diagonal, resulting in an *improper* curve (Dorfman et al., 1997). This may reflect that there exists a non-monotone relationship between the marker and the response with a potential classification use. In this case, we have to define classification subsets different from standard $s_t = (c_t, \infty)$, and the use of the marker becomes more complex.

1.2.1 Generalized ROC curve: the gROC curve

With the aim of accommodating scenarios where both higher and lower values of the marker are associated with a higher risk of having the characteristic, Martínez-Camblor et al. (2017) proposed the so-called *generalized ROC (gROC) curve*. This curve tracks the highest sensitivity for every specificity in the unit interval resulting from subsets of the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$ with $x_t^L \leq x_t^U \in \mathcal{S}(X)$.

The *generalized ROC (gROC) curve* may be defined as any of the following equivalences:

$$\begin{aligned}\mathcal{R}_g(t) = \sup_{s_t \in \mathcal{I}_g(t)} \mathbb{P}(\xi \in s_t) &= \sup_{\substack{x_t^L \leq x_t^U \in \mathcal{S}(X): \\ F_\chi(x_t^U) - F_\chi(x_t^L) = 1-t}} \left\{ F_\xi(x_t^L) + 1 - F_\xi(x_t^U) \right\} \\ &= \sup_{\gamma \in [0,1]} \left\{ F_\xi(F_\chi^{-1}(\gamma t)) + 1 - F_\xi(F_\chi^{-1}(1-[1-\gamma]t)) \right\} \\ &= \sup_{\gamma \in [0,1]} \left\{ \mathcal{R}_r(\gamma t) + 1 - \mathcal{R}_r(1-[1-\gamma]t) \right\} \quad , \ t \in [0,1]\end{aligned}$$

where $\mathcal{I}_g(t) = \left\{ s_t = (-\infty, x_t^L] \cup (x_t^U, \infty) : x_t^L \leq x_t^U \in \mathcal{S}(X) \ , \ \mathbb{P}(\chi \in s_t) = t \right\}$ is the *both-sided family of eligible classification subsets*.

The two latest equivalences are based on the following redefinition of eligible x_t^L and x_t^U in terms of t:

$$x_t^L = F_\chi^{-1}(\gamma t) \qquad \text{and} \qquad x_t^U = F_\chi^{-1}\left(1-[1-\gamma]t\right) \quad \text{with } \gamma \in [0,1].$$

It becomes crucial to consider the supremum in the definition of the gROC curve because the decision rule for each $t \in [0,1]$ is not univocally defined: there exist infinite pairs $x_t^L \leq x_t^U$ reporting a specificity $1-t$ (i.e. $\mathcal{I}_g(t)$ is uncountably infinite). Among those, only the one reporting the maximum sensitivity is considered.

Despite final decisions are based on the underlying classification subsets, they are typically not represented. This omission is not considered a shortcoming in standard cases, as for each specificity $1-t \in [0,1]$, both the associated sensitivity and the subset $s_t = (c_t, \infty)$ are univocally defined by $Se(c_t) = \mathbb{P}(\xi \in s_t) = 1 - F_\xi(c_t)$ and $c_t = 1 - F_\chi^{-1}(1-t)$, respectively. There is only one classification rule of this form resulting in a fixed specificity $1-t$ (same consideration without loss of generality if we fix a sensitivity). Nevertheless, if the gROC curve is considered, given a particular specificity $1-t \in [0,1]$, there are infinite subsets in the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$ satisfying that $\mathbb{P}(\chi \in s_t) = t$. This loss of univocity underlines the importance of reporting (numerically and/or graphically) the decision rules actually proposed for doing the classification. This gap is covered in the presented package. Particularly, the implemented `gROC()` function estimates the gROC curve, both in the studied direction (`side = "both"`) and in the opposite, that is, when classification subsets of the form $s_t = (x_t^L, x_t^U]$ are considered (`side = "both2"`). This last one is defined as

$$\mathcal{R}_{g'}(t) = \sup_{s_t \in \mathcal{I}_{g'}(t)} \mathbb{P}(\xi \in s_t) = \sup_{\substack{x_t^L \leq x_t^U \in \mathcal{S}(X): \\ F_\chi(x_t^U) - F_\chi(x_t^L) = t}} \left\{ F_\xi(x_t^U) - F_\xi(x_t^L) \right\}$$

where $\mathcal{I}_{g'}(t) = \left\{ s_t = (x_t^L, x_t^U] : x_t^L \leq x_t^U \in \mathcal{S}(X) \ , \ \mathbb{P}(\chi \in s_t) = t \right\}$.

Self-contained subsets condition By considering this generalization, another property of the classification subsets may be lost: the classification regions may not be self-contained over the increase in false-positive rate. It may happen that a subject is classified as a positive for a particular FPR t_1, but as a negative for a higher FPR t_2. Therefore, it is natural to establish the following restriction on the classification subsets:

(C) Let $t_1, t_2 \in [0,1]$ with $t_1 \leq t_2$ be the corresponding eligible subsets such that $\mathcal{R}_g(t_1) = \mathbb{P}(\xi \in s_{t_1})$ and $\mathcal{R}_g(t_2) = \mathbb{P}(\xi \in s_{t_2})$, then $s_{t_1} \subseteq s_{t_2}$.

This condition means that, any subject classified as a positive for a fixed specificity (or sensitivity) will be also classified as a positive for any classification subset with lower specificity (higher sensitivity). Pérez-Fernández et al. (2021) proposed an algorithm to estimate the gROC curve under restriction *(C)*, included in the `gROC()` function of the presented R package. It should be noted that this method involves a quite exhaustive search of the self-contained classification subsets leading to the optimal gROC curve estimate. It benefits from the `allShortestPaths()` function in the **e1071** package (Meyer et al., 2023), which implements the Floyd's algorithm (Floyd, 1962) for finding shortest paths in a directed weighted graph. However, even selecting different false-positive rates t_0 to start from, it may not result in the optimal achievable estimate under restriction *(C)*. Input parameters `restric`, `optim`, `t0` and `t0max` for `gROC()` function serve to control this search.

Binormal scenario To accommodate those heteroscedastic binormal scenarios where $\xi \sim \mathcal{N}(\mu_\xi, \sigma_\xi)$ and $\chi \sim \mathcal{N}(\mu_\chi, \sigma_\chi)$ with different $\sigma_\xi \neq \sigma_\chi$, the gROC curve is more appropriate. Assuming that

$\sigma_\xi \geq \sigma_\chi$, the *binormal gROC curve* (Martínez-Camblor and Pardo-Fernández, 2019) is

$$\mathcal{R}_g^{\mathcal{N}}(t) = \sup_{\gamma\in[0,1]} \left\{ \Phi\left(a + b\,\Phi^{-1}(\gamma t)\right) + 1 - \Phi\left(a + b\,\Phi^{-1}\left(1-[1-\gamma]t\right)\right) \right\}\ ,\ t \in [0,1],$$

where $a = \left(\mu_\xi - \mu_\chi\right)/\sigma_\xi$ and $b = \sigma_\chi/\sigma_\xi$ measure the difference in means and variances, respectively. By definition, $\mathcal{R}_g^{\mathcal{N}}(0) = 0$ and $\mathcal{R}_g^{\mathcal{N}}(1) = 1$, while for each $t \in [0,1]$, $\gamma_t \in [0,1]$ reporting the supremum of the sensitivity is the solution to

$$\gamma_t = \frac{1}{t}\,\Phi\left(\frac{2ab}{b^2-1} - \Phi^{-1}\left(1-[1-\gamma_t]t\right)\right).$$

The cut-off values x_t^L and x_t^U defining the classification subsets underlying the binormal gROC curve, $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$, are symmetrical about the *central value* $\mu^* = \dfrac{b^2\mu_\xi - \mu_\chi}{b^2-1}$ for every $t \in (0,1)$. Consequently, the binormal gROC curve fulfills the aforementioned restriction *(C)*.

Assuming the binormal model, the `gROC_param()` function estimates the binormal standard ROC and gROC curve, $\mathcal{R}_r^{\mathcal{N}}(\cdot)$ and $\mathcal{R}_g^{\mathcal{N}}(\cdot)$, respectively, by considering the sample mean estimates ($\overline{\xi_n}$, $\overline{\chi_m}$), and standard deviation estimates ($\hat{s}_\chi$, $\hat{s}_\xi$).

1.2.2 Efficient ROC curve: the eROC curve

By keeping classification subsets of the form $s_t = (c_t, \infty)$, an alternative approach can be explored: transforming the marker through a suitable function $h(\cdot)$ to enhance its accuracy. The concept is that the right-sided ROC curve of the transformed marker $h(X)$ effectively captures the differences between populations better than original marker X. The ROC curve resulting from a transformation $h(\cdot)$ is defined by $\mathcal{R}_h(t) = 1 - F_{h(\xi)}\left(1 - F^{-1}_{h(\chi)}(1-t)\right)$ for $t \in [0,1]$. Kauppi (2016) denoted by *efficient ROC curve* the one resulting from the optimal functional transformation in the ROC sense among all the possible ones. Recall that the standard ROC curve is invariant under monotone transformations of the marker, that is, if $h(\cdot)$ is a monotone increasing function, $\mathcal{R}_h(\cdot) = \mathcal{R}_r(\cdot)$, while if it is decreasing, $\mathcal{R}_h(\cdot) = \mathcal{R}_l(\cdot)$.

Here we revisit how to improve the binary classification performance of univariate markers via functional transformations and its impact on the final classification regions. With this goal, the `hROC()` function and others related (`plot_regions()` and `plot_funregions()`) were implemented. Depending on the monotonicity of the function $h(\cdot)$, the resulting classification subsets for X have different shapes, despite the rules for $h(X)$ are of the form $s_t = (c_t, \infty)$.

Henceforth, the transformation $h^*(\cdot)$ reporting the dominant ROC curve compared to the ROC curve from any other transformation $h(\cdot)$ (i.e. $\mathcal{R}_{h^*}(\cdot) \geq \mathcal{R}_h(\cdot)$) will be referred to as *optimal transformation* (in the ROC sense). With the purpose of estimating such optimal transformation, two different semiparametric procedures are reviewed:

1.- Martínez-Camblor et al. (2019) exploited the result proved by McIntosh and Pepe (2002), suggesting to estimate the logit of the risk function by logistic regression. That is,

$$h^*(\cdot) = \text{logit}\left\{\mathbb{P}(D = 1 \mid \cdot)\right\}$$

where D is the random variable modeling the population the subject belongs to ($D = 1$ if the subject has the characteristic of interest, $D = 0$ otherwise). This function is a monotone increasing transformation of the likelihood ratio function, $\mathcal{L}(\cdot) = f_\xi(\cdot)/f_\chi(\cdot)$, which is the optimal one in the ROC sense by the Neyman-Pearson lemma. To clarify notation, $f_\xi(\cdot)$ and $f_\chi(\cdot)$ are the density functions for the positive and the negative population, respectively. Namely,

$$\text{logit}\left\{\mathbb{P}(D = 1 \mid X)\right\} = \text{logit}\left\{\frac{\mathcal{L}(X)\cdot q}{1 + \mathcal{L}(X)\cdot q}\right\} \qquad \text{where } q = \frac{\mathbb{P}(D=1)}{\mathbb{P}(D=0)}.$$

By the proposed R tool, the user can define any transformation $h(\cdot)$ for the right-hand side of the logistic regression model to be fitted, $\text{logit}\left\{\mathbb{P}(D = 1 \mid x)\right\} = h(x)$, to estimate the optimal transformation. Particularly, by fixing `type = "lrm"` as an input parameter for the `hROC()` function, the user may define any function $h(\cdot)$ by the input `formula.lrm`. The user can make use of special transformation functions such as `pol()` (polynomial) or `rcs()` (restricted cubic splines) by loading the library **rms**.

2.- Arguing as in Martínez-Camblor et al. (2021b) for univariate markers instead of multivariate,

the optimal transformation in the ROC sense is equivalent to

$$h^*(\cdot) = \frac{f_\xi(\cdot)}{f_\xi(\cdot) + f_\chi(\cdot)}.$$

The sum in the denominator is applied to avoid zeros. In order to estimate $h^*(\cdot)$, different estimation procedures for the density functions separately may be used, such as kernel density estimator. The main handicap of the latter is that it is based on a bandwidth chosen by the user. By our package, by fixing `type = "kernel"` as an input parameter for the `hROC()` function and choosing a bandwidth by the input `kernel.h` (1 by default), the user may compute this method.

3.- Martínez-Camblor et al. (2019) also argues that *with no restrictions on the shape of $h^*(\cdot)$ and based on a fixed sample of positive and negative subjects without ties between the positive and negative values, it is always possible to build a function which provides a perfect classification region*. This is called the *overfitting function* and its estimation (since it totally relies on the available sample) is defined as follows:

$$\widehat{h^*_{of}}(x) = \sum_{i=1}^{n_1} I(x = y_i) + \sum_{i=1}^{n_2} \frac{\#(\xi = z_i)}{\#(\xi = z_i) + \#(\chi = z_i)} I(x = z_i)$$

where $I(A)$ denotes the indicator function (which takes the value 1 if A is true and 0 otherwise), $\#(B)$ is the cardinal of the subset B, $\{y_1, \ldots, y_{n_1}\} \subseteq \{\xi_1, \ldots, \xi_n\}$ are the positive sample values without ties and $\{z_1, \ldots, z_{n_2}\} \subseteq \{\xi_1, \ldots, \xi_n\}$ are the positive sample values with ties with any negative sample value. Classification based on this transformation is the optimal one in the AUC sense, but the resulting decision rules cannot be extended to any other sample.
By the `hROC()` function of our package, this transformation may be estimated by fixing the input parameter `type` to `"overfitting"`.

Relationship with the gROC curve Martínez-Camblor et al. (2021a) proved that, under restriction *(C)*, the gROC curve based on the classification subsets $\left\{s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)\right\}_{t\in[0,1]}$ is equivalent to the right-sided ROC curve of the transformed marker $h_{gC}(X)$ by the following function:

$$h_{gC}(x) = \int_0^1 I(x < x_t^L)dt + \int_0^1 I(x > x_t^U)dt \ , x \in \mathcal{S}(X),$$

where $I(A)$ denotes the indicator function, which takes the value 1 if A is true and 0 otherwise. This relationship is used in the implementation of `plot_funregions()` function for `` `groc` `` objects, which is only allowed when restriction *(C)* is fulfilled (self-contained classification subsets). Otherwise, such transformation does not exist, as argued in Pérez-Fernández et al. (2021) (Appendix).

1.3 Multivariate ROC curve

We want to point out that the ROC curve is defined for classification accuracy evaluation of univariate markers. When dealing with multivariate markers ($\boldsymbol{X} \in \mathbb{R}^p$), the usual practice is to consider a transformation $\boldsymbol{h} : \mathbb{R}^p \longrightarrow \mathbb{R}$, usually linear, to reduce it to a univariate marker, and then to construct the standard ROC curve. Same considerations as before apply when a functional transformation is taken. In the proposed R library, we consider several methods to define such transformation in the multivariate scenario, from the existing literature, most of them listed and mathematically explained in Pérez-Fernández et al. (2021) and Martínez-Camblor et al. (2021b).

The function of the **movieROC** package dealing with multivariate markers is the `multiROC()` function. It considers one of these methods, chosen by the user by the input parameter `method`:

i) fitting a binary logistic regression model with a particular combination of the components on the right-hand side (`method = "lrm"`),

ii) linear combinations with fixed parameters (`method = "fixedLinear"`),

iii) linear combinations with dynamic parameters (`method = "dynamicMeisner"` or `method = "dynamicEmpirical"`),

iv) quadratic combinations with fixed parameters (`method = "fixedQuadratic"`), or

v) estimating the optimal transformation by kernel density techniques deeply studies in Duong (2007) (`method = "kernelOptimal"`).

The latter two are only available for bivariate markers. Dealing with bivariate markers has been much more exploited in the scientific literature. Some extensions are implemented to deal with multivariate markers when $p > 2$, mainly by using linear combinations thus far.

2 Main functions of the movieROC package and illustrative dataset

To enhance the comprehension of the developed R package, Section 2.1 provides a detailed description of the main objectives of the implemented R functions. Furthermore, to reflect its practical usage, we employ a real dataset throughout this vignette, which is introduced in Section 2.3.

2.1 Functionality of the movieROC package

A graphical tool was developed to showcase static and dynamic graphics displaying the classification subsets derived from maximizing diagnostic accuracy under certain assumptions, ensuring the preservation of the interpretability. The R package facilitates the construction of the ROC curve across various specificities, providing visualizations of the resulting classification regions. The proposed tool comprises multiple R functions that generate objects with distinct class attributes (see function names where red arrows depart from and red nodes in Figure 1, respectively). Once the object of interest is created, different functions may be passed to them, in order to plot the underlying classification regions (`plot_regions()`, `plot_funregions()`), to track the resulting ROC curve (`plot_buildROC()`, `plot()`), to `predict` decision rules for a particular specificity, and to `print` relevant information, among others. The main function of the package, `movieROC()`, produces videos to exhibit the classification procedure.

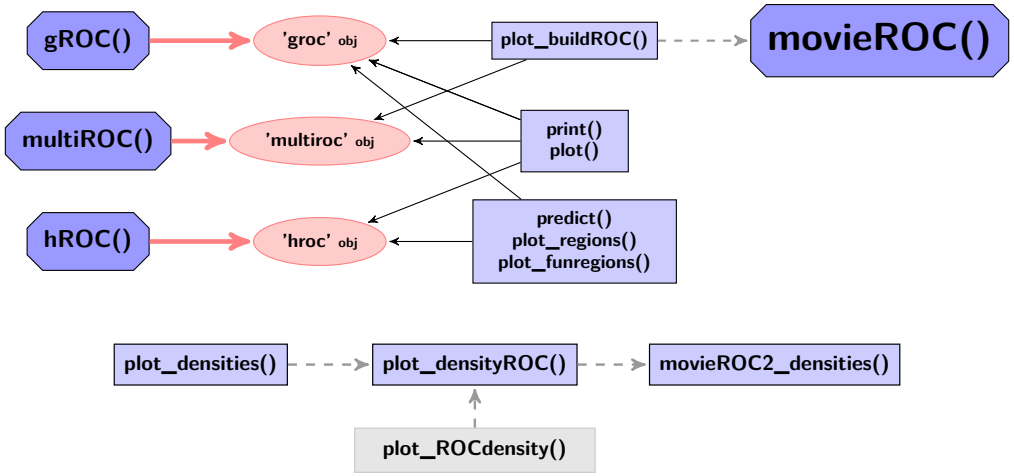


Figure 1: R functions of the **movieROC** package. The blue nodes include the names of the R functions and the red nodes indicate the different R objects that can be created and worked with. The red arrows depart from those R functions engaged in creating R objects and the black arrows indicate which R functions can be applied to which R objects. The grey dashed arrows depart from those R functions internally used in the implementation of the R function at the end of the arrow.

2.2 Class methods for movieROC objects

By using the `gROC()`, the `multiROC()` or the `hROC()` function, the user obtains an R object of class 'groc', 'multiroc' or 'hroc', respectively. These will be called **movieROC** objects.

Following tables provides the main input and output parameters of those function (dark blue nodes at top diagram of Figure 1). Namely,

Table 1: `gROC()` and the `gROC_param()` functions, which estimate the regular and the generalized ROC curve and associated decision rules, by the empirical or the binormal approach, respectively;
MAIN SYNTAX: `gROC(X,D,side = "right",...)` `gROC_param(X,D,side = "right",...)`

Table 2: `multiROC()` function, which estimates the ROC curve resulting from a multivariate marker;
MAIN SYNTAX:
```
multiROC(X,D,method = "lrm",
  formula = 'D ~ X.1 + I(X.1^2) + X.2 + I(X.2^2) + I(X.1*X.2)' ,...)
```

Table 3: `hROC()` function, which estimates the ROC curve resulting from a suitable functional transformation of a univariate marker;
MAIN SYNTAX: `hROC(X,D,type = "lrm",formula = 'D ~ pol(X,3)' ,...)`

Table 1: Most relevant parameters of the `gROC()` function. The `gROC_param()` function works similarly when the binormal scenario is assumed.

Input parameters	
`X`	Vector of marker values.
`D`	Vector of response values. Two levels; if more, the two first ones are used.
`side`	Type of ROC curve. One of `"right"` ($\mathcal{R}_r(\cdot)$), `"left"` ($\mathcal{R}_l(\cdot)$), `"both"` ($\mathcal{R}_g(\cdot)$) or `"both2"` ($\mathcal{R}_{g'}(\cdot)$). Default: `"right"`.
`N`	Length of the vector of FPR used to build the ROC curve: $t \in \{0, 1/N, 2/N, \ldots, 1\}$. Default: `1000`. Only used for the `gROC_param()` function.
`restric`	If TRUE, the gROC curve with restriction *(C)* is computed. Default: FALSE.
`optim`	If TRUE (and `restric = TRUE`), the computation of the optimal gROC curve under restriction *(C)* is performed by using Floyd's algorithm (Floyd, 1962). It is computed by the `allShortestPaths()` function in the **e1071** package. Default: TRUE.
`t0`	An integer number between 1 and $m+1$. If `restric = TRUE`, the restricted gROC curve is computed departing from (`t0`$-1)/m$. Default: the one reporting the Youden index.
`t0max`	If TRUE, the computation of the gROC curve under restriction *(C)* is performed departing from every possible `t0` and the one reporting the maximum AUC is selected.

Output parameters	
`controls,cases`	Marker values of negative and positive subjects, respectively.
`t`	Vector of false-positive rates.
`roc`	Vector of values of the ROC curve for `t`.
`c`	Vector of marker thresholds resulting in (`t`, `roc`) if `side = "right" | "left"`.
`xl,xu`	Vectors of marker thresholds resulting in (`t`, `roc`) if `side = "both" | "both2"`.
`auc`	Area under the curve estimate.
`a,b`	Estimates for parameters a and b used in ROC curve estimation: $\hat{a} = \left[\overline{\xi_n} - \overline{\chi_m}\right] / \hat{s}_\xi$, $\hat{b} = \hat{s}_\chi / \hat{s}_\xi$. Only used for the `gROC_param()` function.
`p0`	Estimate of the 'central value', μ^*, about to which the thresholds x^L and x^U are symmetrical. Only used for the `gROC_param()` function.
`aucfree`	Area under the curve estimate without restrictions.
`aucsi0`	gAUC under restriction *(C)* departing from every false-positive rate, $FPR \in \{0, 1/m, \ldots, 1\}$.

Table 2: Most relevant input and output parameters of the `multiROC()` function.

Input parameters	
`X`	Matrix of marker values with dimension $N \times p$.
`D`	Vector of response values. Two levels; if more, the two first ones are used.
`method`	Method used to build the classification regions. One of `"lrm"` (fitting a binary logistic regression model by the input parameter `formula` computed by using the `lrm()` function in the **rms** package), `"fixedLinear"` (linear frontiers with fixed parameters given in `coefLinear` or estimated by method in `methodLinear`), `"fixedQuadratic"` (quadratic frontiers with fixed parameters given in `coefQuadratic`, only available for $p = 2$), `"dynamicMeisner"` (linear frontiers with dynamic parameters reported by Meisner et al. (2021) method computed by using the `maxTPR()` function in the **maxTPR** package), `"dynamicEmpirical"` (linear frontiers with dynamic parameters reported by the empirical method, only available for $p = 2$), `"kernelOptimal"` (estimating optimal transformation based on bivariate kernel density estimation by using the `kde()` function in the **ks** package). Default: `"lrm"`.
`formula.lrm`	If `method = "lrm"`, the transformation employed in the right-hand side of the logistic regression model (in terms of `X.1`, `X.2`, ..., `X.p`, and `D`). Default: `'D ~ X.1 + I(X.1^2) + X.2 + I(X.2^2) + I(X.1*X.2)'`.
`stepModel`	If TRUE and `method = "lrm"`, a model selection is performed based on the AIC (Akaike information criterion) in a stepwise algorithm (by `stats::step()` function). Default: TRUE.
`methodLinear`	If `method = "fixedLinear"`, method used to estimate the coefficients β_i ($i \in \{1, \ldots, p\}$) for $\mathcal{L}_{\boldsymbol{\beta}}(\boldsymbol{X}) = \beta_1 X_1 + \cdots + \beta_p X_p$. One of `"coefLinear"` (particular fixed coefficients in `coefLinear`), `"SuLiu"` (Su and Liu, 1993), `"PepeThompson"` (Pepe and Thompson, 2000), `"logistic"` (logistic regression model), or `"minmax"` (Liu et al., 2011). Default: `"coefLinear"`.

`coefLinear`	If `method = "fixedLinear"`, a vector of length p with the coefficients β_i ($i \in \{1,\ldots,p\}$) used in $\mathcal{L}_{\boldsymbol{\beta}}(\boldsymbol{X}) = \beta_1 X_1 + \cdots + \beta_p X_p$. Default: $(1,\ldots,1)$.
`coefQuadratic`	If `method = "fixedQuadratic"`, a vector of length 5 with the coefficients $\beta_1,\ldots,\beta_5$ used in $\mathcal{Q}_{\boldsymbol{\beta}}(\boldsymbol{X}) = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2$. Default: $(1,1,0,1,1)$. Only for $p = 2$.
`alpha,approxh, multiplier`	If `method = "dynamicMeisner"`, input parameters used in the `maxTPR()` function in the **maxTPR** package. Default: `alpha = 0.5`, `approxh = 0.5` and `multiplier = 2`.
`K`	If `method = "dynamicEmpirical"`, the number of equally spaced $\alpha \in (-1,1)$ studied. Default: 201. Only available for $p = 2$.
`kernelOptimal.H`	If `method = "kernelOptimal"`, the estimation procedure for the bandwidth (Duong, 2007). Default: `"Hbcv"` (biased cross-validation bandwidth matrix) if $p = 2$ and `"Hpi"` (plug-in bandwidth selector) if $p > 2$.

Output parameters	
`controls,cases`	Marker values of negative and positive subjects, respectively.
`Z`	If `method = "lrm" | "fixedLinear" | "fixedQuadratic" | "kernelOptimal"`, resulting univariate marker values. `"fixedQuadratic"` only available for $p = 2$.
`c`	If `method = "lrm" | "fixedLinear" | "fixedQuadratic" | "kernelOptimal"`, vector of final marker thresholds resulting in (`t`, `roc`). `"fixedQuadratic"` only available for $p = 2$.
`CoefTable`	If `method = "dynamicMeisner" | "dynamicEmpirical"`, a list of length equal to length of vector `t`. Each element of the list keeps the linear coefficients (`coef`), threshold for such linear combination (`c`), the corresponding point in the ROC curve (`t`, `roc`), the resulting univariate marker values (`Z`) and a matrix of dimension 100×100 with the marker values over a grid of (X_1, X_2) bivariate values (`f`). Last object only available for $p = 2$.
`t`	Vector of false-positive rates.
`roc`	Vector of values of the ROC curve for `t`.
`auc`	Area under the curve estimate.

Table 3: Most relevant input and output parameters of the `hROC()` function.

Input parameters	
`X`	Vector of marker values.
`D`	Vector of response values. Two levels; if more, the two first ones are used.
`type`	Type of transformation considered. One of `"lrm"` (a binary logistic regression is computed by using the `lrm()` function in the **rms** package, Harrell Jr (2023)), `"kernel"` (the transformation included in the second method above estimated by kernel density approach), `"overfitting"` (the overfitting transformation, $h_{of}(\cdot)$ is taken), or `"h.fun"` (the transformation indicated in the input parameter `h.fun` is considered). Default: `"lrm"`.
`formula.lrm`	If `type = "lrm"`, the transformation employed in the right-hand side of the logistic regression model (in terms of `X` and `D`). Default: `'D ~ pol(X, 3)'`.
`kernel.h`	If `type = "kernel"`, the bandwidth used for the kernel density estimation by using the `density()` function in the **stats** package. Default: 1.
`h.fun`	If `type = "h.fun"`, the transformation employed (as a function in R). Default: `function(x){x}`.
`plot.h`	If TRUE, the transformation employed is illustrated.
`plot.roc`	If TRUE, the resulting ROC curve is illustrated.

Output parameters	
`Y`	Transformed marker values.
`Sp,Se`	Vector of true-negative and true-positive rates, respectively.
`auc`	Area under the curve estimate.
`model`	If `type = "lrm"`, the coefficients of the logistic regression model fitted by `formula`.

Once the object of interest is created, many functions may be passed to it. Some of them are generic methods (`print()`, `plot()` and `predict()`), commonly used in R language over different objects according to their class attributes. The rest of the functions are specific for this library and only applicable to **movieROC** objects. Table 4 summarizes the functions `plot_densities()` and `plot_densityROC()` and provides their target and main syntax. The main methods implemented (light blue nodes at top diagram of Figure 1) are explained in the main manuscript of the package.

Table 4: Brief explanation of `plot_densities()` and `plot_densityROC()` function usage. Both can be applied to a 'groc' object (output of the `gROC()` function). The main input parameters are displayed.

Specific functions	
`plot_densities()`	Applicable only to a 'groc' object. Plot the density function estimates for the marker distribution in positive and negative populations (in red and blue color by default, respectively). MAIN SYNTAX: `plot_densities(obj,h = c(1,1),histogram = FALSE,breaks = 15,...)` By default, the density functions are estimated by `stats::density()` function with input parameter `adjust = h[i]` for $i \in \{1 = controls, 2 = cases\}$. Instead, histograms may be reported by means of `graphics::hist()` function with a fixed number of `breaks` (15 by default). `obj` is the output of `gROC()` function, but parameter `side` is not considered.
`plot_densityROC()`	Applicable only to a 'groc' object. Plot the ROC curve estimate based on kernel density estimation functions for the marker distribution in both positive and negative populations. MAIN SYNTAX: `plot_densityROC(obj,h = c(1,1),C = NULL,build.process = FALSE,...)` The classification procedure may be displayed for a cut-off value `C` introduced by the user. Note: This function is only valid for right-sided and left-sided ROC curves.

2.3 Illustrative dataset. `plot_densities()` function

In order to illustrate functionality of our R package, we consider the `HCC` data. This dataset is derived from gene expression arrays of tumor and adjacent non-tumor tissues of 62 Taiwanese cases of hepatocellular carcinoma (HCC). The goal of the original study (Shen et al., 2012) was to identify, with a genome-wide approach, additional genes hypermethylated in HCC that could be used for more accurate analysis of plasma DNA for early diagnosis, by using Illumina methylation arrays (Illumina, Inc., San Diego, CA) that screen 27,578 autosomal CpG sites. The complete dataset was deposited in NCBI's Gene Expression Omnibus (GEO) and it is available through series accession number GSE37988 (www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37988). It is included in the presented package (`HCC` dataset), excluding 275 genes with incomplete information and selecting 948 among remaining.

The following code loads the R package and the `HCC` dataset and shows its main structure:

```
R> library(movieROC)
R> data(HCC)
R> str(HCC)
```

```
'data.frame':        124 obs. of  952 variables:
 $ tissue    : num  1 2 3 4 5 6 7 8 9 10 ...
 $ sex       : chr  "male" "male" "male" "female" ...
 $ age       : num  NA 67 47 70 NA NA 56 NA 39 78 ...
 $ tumor     : chr  "tumor" "tumor" "tumor" "tumor" ...
 $ cg03409548: num  0.0495 0.0315 0.0752 0.0433 0.0559 ...
 $ cg03410718: num  0.93 0.913 0.943 0.94 0.948 ...
 $ cg03411288: num  0.025 0.0146 0.0226 0.0271 0.0319 ...
  (...)
 $ cg20239949: num  0.0332 0.0292 0.0376 0.0328 0.0679 ...
 $ cg20240860: num  0.0822 0.07 0.0502 0.1175 0.2827 ...
```

```
R> table(HCC$tumor)
```

```
nontumor    tumor
      62       62
```

We selected the genes 20202438, 18384097, and 03515901. On the one hand, we chose the gene 03515901 as an example of a monotone relationship between the marker and the response, reporting a good left-sided ROC curve. On the other hand, relative gene expression intensities of the genes 20202438 and 18384097 tend to be more extreme in tissues with tumor than in those without it. These are non-standard cases, so if we limit ourselves to detect "appropriate" genes on the basis of the standard ROC curve, they would not be chosen. However, extending the decision rules by means of the gROC curve, those genes may be considered as potential biomarkers (locations) to differ between the two groups.

The next chunk of code summarizes the distribution of the three selected genes, separated by tumor or non-tumor tissues.

```
R> genes <- c("20202438", "18384097", "03515901")
R> summary_genes <- lapply(genes, function(gene){
+   by(HCC[,paste0("cg",gene)], HCC$tumor, summary)
+ })
R> names(summary_genes) <- paste("Gene", genes)
R> summary_genes

$`Gene 20202438`
HCC$tumor: nontumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6025  0.7549  0.7724  0.7665  0.7930  0.8356
------------------------------------------------------------
HCC$tumor: tumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4849  0.7245  0.7828  0.7610  0.8166  0.8683

$`Gene 18384097`
HCC$tumor: nontumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.3952  0.6294  0.6630  0.6609  0.7086  0.8008
------------------------------------------------------------
HCC$tumor: tumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0268  0.6407  0.7628  0.7169  0.8354  0.8952

$`Gene 03515901`
HCC$tumor: nontumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4291  0.6713  0.7206  0.7188  0.7755  0.8793
------------------------------------------------------------
HCC$tumor: tumor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1662  0.3588  0.4752  0.5096  0.6732  0.9472
```

The next piece of code represents the density estimation for gene expression intensities of the selected genes in each group, shown in Figure 2, using the `plot_densities()` function.

```
R> par(mfrow = c(1,3))
R> for(gene in c("20202438", "18384097", "03515901")){
+   roc <- gROC(X = HCC[,paste0("cg",gene)], D = HCC$tumor)
+   plot_densities(roc, histogram = TRUE, lwd = 3, main = paste("Gene", gene),
+                  legend = (gene == "03515901"), pos.legend = "topleft",
+                  xlim = c(0.4*(gene == "20202438"),1))
+   plot_densities(roc, lwd = 3, new = FALSE,
+                  col = adjustcolor(c('#485C99','#8F3D52'), alpha.f = 0.8))}
```

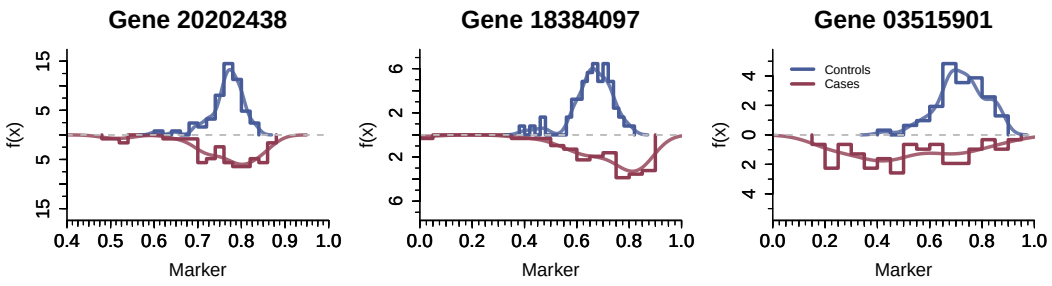


Figure 2: Density histograms and kernel density estimations (lighter) for gene expression intensities of the genes 20202438, 18384097 and 03515901 in negative (non-tumor) and positive (tumor) tissues.

3 Study of the classification performance of gene 20202438 by using the movieROC package

For the next subsections (except the last one, multivariate), the marker used will be the gene expression intensity of the gene 20202438 and the goal is to classify tissues in tumor or non-tumor.

```
R> X <- HCC$cg20202438; D <- HCC$tumor
```

3.1 The regular ROC curve. `gROC()` and `predict()` functions

The function `gROC()` performs an empirical estimation of the right-sided ROC curve by default.

```
R> roc <- gROC(X, D)
R> roc
```

```
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that larger values of the marker indicate larger confidence that a
 given subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.855 and 0.403,
 respectively, corresponding to the following classification subset: (0.799, Inf).
The area under the right-sided ROC curve (AUC) is 0.547.
```

With the function `predict()`, we can numerically display the classification subset and sensitivity resulting in a false-positive rate of 0.1.

```
R> predict(roc, FPR = .1)
```

```
$ClassSubsets                          $Specificity               $Sensitivity
[1] 0.8063487       Inf                [1] 0.9032258              [1] 0.3064516
```

3.2 The gROC curve. `gROC()` and `predict()` functions

The function `gROC()` with input `side = "both"` performs an empirical estimation of the gROC curve without restrictions. Now the decision rules are of the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$.

```
R> groc <- gROC(X, D, side = "both")
R> groc
```

```
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that both lower and larges values of the marker indicate larger
 confidence that a given subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.694 and 0.726,
  respectively, corresponding to the following classification subset:
  (-Inf, 0.737) U (0.799, Inf).
The area under the gROC curve (gAUC) is 0.765.
```

With the function `predict()`, we can numerically display the classification subset and sensitivity resulting in a false-positive rate of 0.1.

```
R> predict(groc, FPR = .1)
```

```
$ClassSubsets                                $Specificity               $Sensitivity
          [,1]      [,2]                     [1] 0.9032258              [1] 0.4032258
[1,]      -Inf 0.7180623
[2,] 0.8296072       Inf
```

3.3 The gROC curve with restriction *(C)*. `gROC()` and `predict()` functions

The function `gROC()` with inputs `side = "both"` and `restric = TRUE` performs an empirical estimation of the gROC curve with restriction *(C)*. Now the decision rules are of the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$ and self-contained along the change of $t \in [0, 1]$. This procedure is time-consuming.

It may be seen that the loss in the gAUC is small with respect to the gROC without restrictions (from 0.765 to 0.762).

```
R> groc_C <- gROC(X, D, side = "both", restric = TRUE)
R> groc_C
```

```
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that both lower and larges values of the marker indicate larger
 confidence that a given subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.694 and 0.726,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.737) U (0.799, Inf).
The area under the gROC curve (gAUC) is 0.762.
```

The classification subset and sensitivity resulting in a false-positive rate of 0.1 is the same than those for the gROC curve without restrictions.

```
R> predict(groc_C, FPR = .1)
```

```
$ClassSubsets                              $Specificity
          [,1]      [,2]                   [1] 0.9032258
[1,]      -Inf 0.7180623
[2,] 0.8296072       Inf                   $Sensitivity
                                           [1] 0.4032258
```

3.4 Some figures for standard and gROC curves. `plot()`, `plot_densityROC()` and `plot_regions()` functions

Right plot in Figure 3 represents the previous empirical estimates in a single figure by the function `plot()` included in **movieROC**. The left panels in Figure 3 come from the kernel estimation of the two probability density functions involved, by using a bandwidth = 1 in the `plot_densityROC()` function. The resulting estimate for standard ROC curve (middle) is very similar to the empirical one (right, black line).

```
R> layout(mat = matrix(1:2, nrow = 1), width = c(1.5,1))
R> plot_densityROC(roc, lwd = 3, par.specify = TRUE)
```

```
R> par(mfrow = c(1,1))
R> plot(roc, main = "Empirical ROC curve", lwd = 3)
R> plot(groc, new = FALSE, lwd = 3, col = "gray50")
R> plot(groc_C, new = FALSE, lwd = 3, lty = 2, col = "blue")
R> legend("bottomright", paste(c("AUC =", "gAUC =", "gAUC_C ="),
+                          format(c(roc$auc, groc$auc, groc_C$auc), digits = 3)),
+         col = c("black", "gray50", "blue"), lwd = 3, lty = c(1,1,2),
+         bty = "n", inset = .01)
```

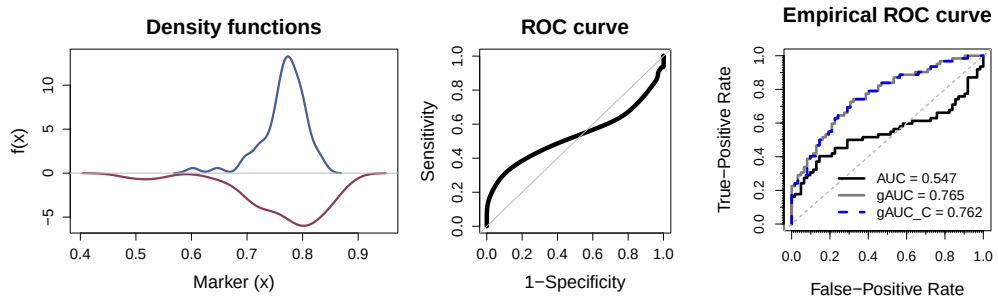


Figure 3: From left to right: i) Kernel density function estimates of gene 20202438 in tumor (in red) and non-tumor (in blue) groups (bandwidth = 1 by default); ii) Standard ROC curve estimate from i); iii) Empirical estimate for standard ROC curve (in black), gROC curve (in gray) and gROC curve with restriction *(C)* (in blue).

The function `plot_densityROC()` also allows to graphically track the construction of the "smooth" ROC curve (by the input parameter `build.process`), displaying the classification subsets until a particular cut-off value chosen by the user (input parameter `C`).

```
R> plot_densityROC(roc, C = .8, build.process = TRUE)
```

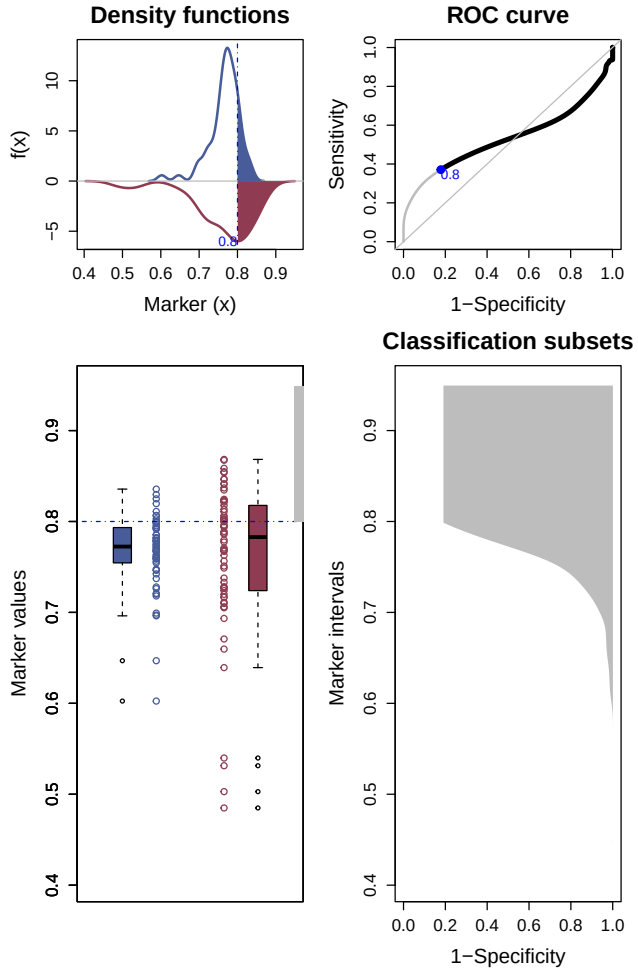
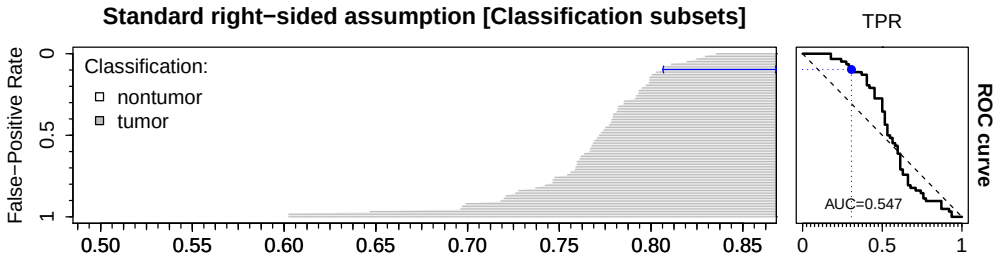


Figure 4: Classification procedure until cut-off value 0.8 for the gene 20202438 by using the `plot_densityROC()` function.

All the classification regions underlying the standard and the generalized ROC curves without and with restrictions are represented in Figure 5. The following code was used to generate the figure, illustrating the usage and output of the `plot_regions()` function. Besides displaying all the classification regions underlying every specificity (in gray), the one chosen by the user (FPR = 0.1 to be consistent with previous sections) is highlighted in blue. Note that the ROC curves are rotated 90° to the right, in order to use the vertical axis for FPR in both plots.

```
R> plot_regions(roc, cex.legend = 1.5, plot.auc = TRUE, FPR = .1,
+                main = "Standard right-sided assumption [Classification subsets]")
R> plot_regions(groc, plot.auc = TRUE, legend = F, main.plotroc = "gROC curve",
+                FPR = .1, main = "General approach [Classification subsets]")
R> plot_regions(groc_C, plot.auc = TRUE, legend = F, main.plotroc = "gROC curve",
+                FPR = .1, xlab = "Gene 20202438 expression intensity",
+              main = "General approach with restriction (C) [Classification subsets]")
```



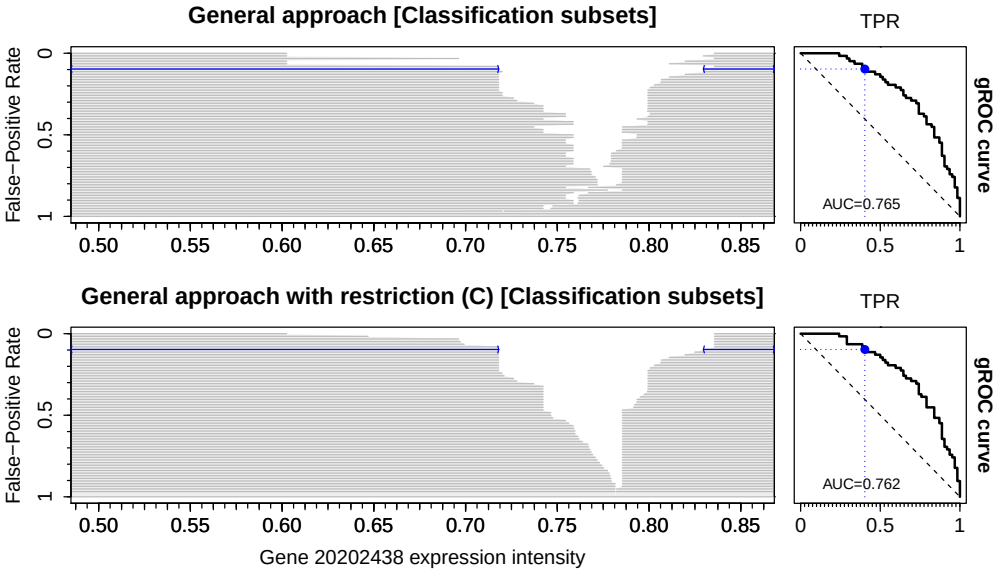


Figure 5: Classification regions and the ROC curve (90º rotated) for evaluation of gene 20202438 expression intensity assuming i) standard scenario (top), ii) generalized scenario without restrictions (middle), iii) generalized scenario under restriction *(C)* over the subsets (bottom).

3.5 Videos for the gROC curve. `movieROC()` and `plot_buildROC()` functions

The following line of code saves a video as a GIF with the construction of the empirical gROC curve:

```
R> movieROC(groc, file = "gROC_gene20202438.gif")
```

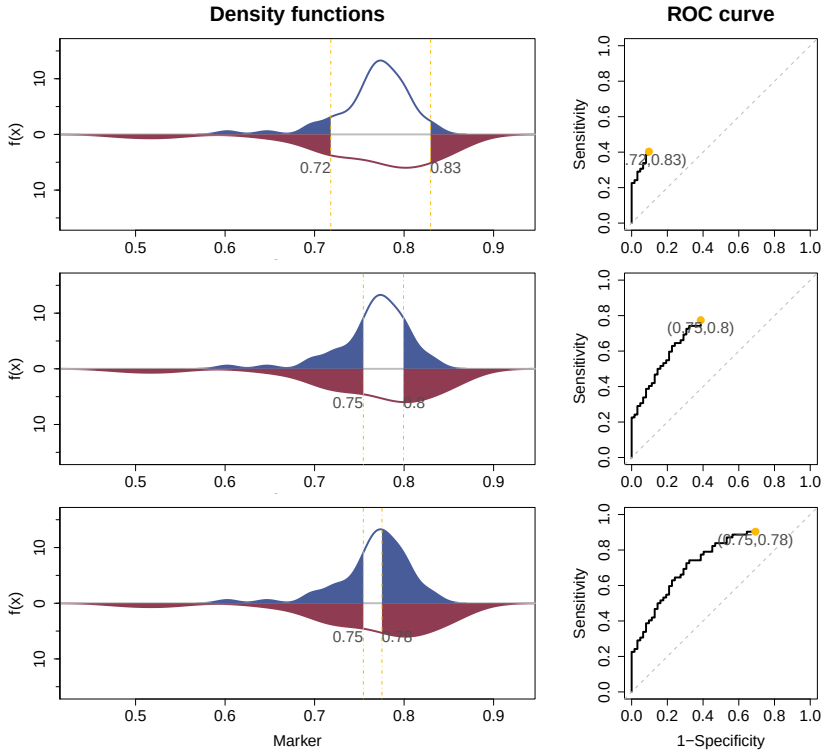


Figure 6: Snapshots (from `movieROC()` function) of the classification procedure and empirical gROC curve for the gene 20202438 when FPR $\in \{0.1, 0.4, 0.7\}$.

These three snapshots can also be generated by the `plot_buildROC()` function, as follows:

```
R> for(fpr in c(.1,.4,.7)) plot_buildROC(groc, FPR = fpr, build.process = TRUE)
```

3.6 The binormal scenario. gROC_param(), plot(), plot_densities and plot_regions() functions

When the binormal scenario is assumed, the results for the regular and gROC curve estimation are the following, obtained by the function gROC_param() in our package:

```
R> norm_roc <- gROC_param(X, D)
R> norm_roc
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that larger values of the marker indicate larger confidence that a given
 subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.923 and 0.226,
 respectively, corresponding to the following classification subset: (0.825, Inf).
The area under the right-sided ROC curve (AUC) is 0.477.

R> norm_lroc <- gROC_param(X, D, side = "left")
R> norm_lroc
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that lower values of the marker indicate larger confidence that a given
 subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.91 and 0.281,
 respectively, corresponding to the following classification subset: (-Inf, 0.712).
The area under the left-sided ROC curve (AUC) is 0.523.

R> norm_groc <- gROC_param(X, D, side = "both")
R> norm_groc
Data was encoded with nontumor (controls) and tumor (cases).
It is assumed that both lower and larges values of the marker indicate larger
 confidence that a given subject is a case.
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.833 and 0.507,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.712) U (0.824, Inf).
The area under the gROC curve (gAUC) is 0.715.
```

Figure 7 illustrates the kernel density estimation for the gene 20202438 in both groups with a high bandwidth and the ROC curve estimates when binormal scenario is assumed. It can be seen that the left-sided ROC curve reports an AUC slightly better than the right-sided ROC curve. However, the biggest gain in AUC is considering the gROC curve, since from the left plot it is clear that the largest differences between the two groups are in deviation, not in location.

```
R> plot_densities(norm_roc, h = 3, lwd = 3, legend = TRUE, pos.legend = "topleft")
R> plot(norm_roc, main = "Binormal ROC curve", lwd = 3)
R> plot(norm_lroc, new = FALSE, lwd = 3, col = "green4")
R> plot(norm_groc, new = FALSE, lwd = 3, col = "gray50")
R> legend("bottomright", paste(c("AUC =", "lAUC =", "gAUC ="),
+                   format(c(norm_roc$auc, norm_lroc$auc, norm_groc$auc), digits = 3)),
+        col = c("black", "green4", "gray50"), lwd = 3, bty = "n", inset = .01)
```

All the classification regions underlying the standard and the generalized ROC curves when binormal scenario is assumed are represented in Figure 8. The following code was used to generate the figure, illustrating the usage and output of the plot_regions() function. Besides displaying all the classification regions underlying every specificity (in gray), the one chosen by the user (FPR = 0.1 to be consistent with previous sections) is highlighted in blue.

```
R> plot_regions(norm_roc, FPR = .1, main.plotroc = "Binormal ROC curve",
+                main = "Standard right-sided assumption  [Classification subsets]",
+                cex.legend = 1.5, plot.auc = TRUE)
R> plot_regions(norm_lroc, FPR = .1, main.plotroc = "Binormal ROC curve",
+                main = "Standard left-sided assumption  [Classification subsets]",
+                legend = FALSE, plot.auc = TRUE)
R> plot_regions(norm_groc, FPR = .1, main.plotroc = "Binormal gROC curve",
+                main = "General approach  [Classification subsets]", legend = FALSE,
+                plot.auc = TRUE, xlab = "Gene 20202438 expression intensity")
```

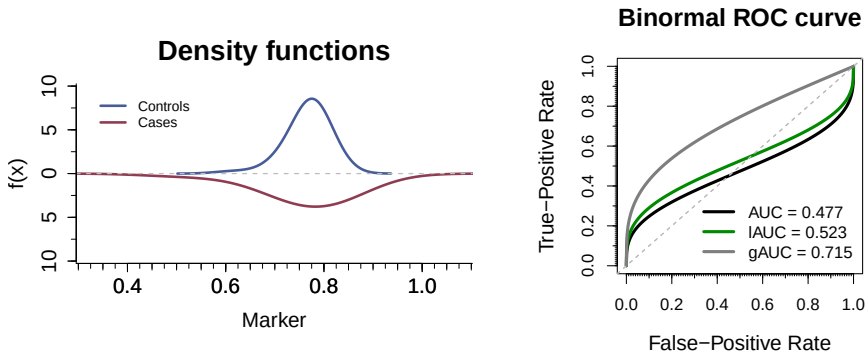


Figure 7: Left, kernel density function estimates of gene 20202438 in tumor (in red) and non-tumor (in blue) groups (bandwidth = 3); right, right- and left-sided ROC curve (in black and green, respectively) and gROC curve (in gray) when the binormal scenario is assumed.

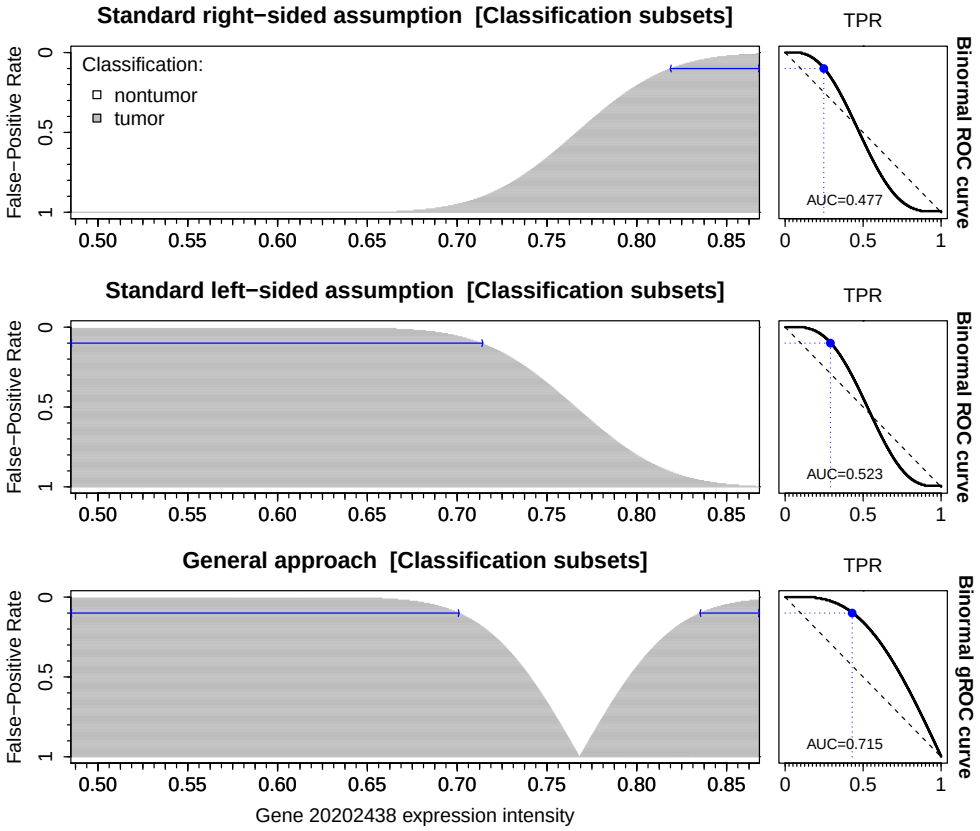


Figure 8: Classification regions and the binormal ROC curve (90º rotated) for gene 20202438 assuming i) standard scenario with subsets of the form $s_t = (c_t, \infty)$ (*right-sided*, top) or $s_t = (-\infty, c_t]$ (*left-sided*, middle), ii) generalized scenario with subsets of the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$ (*both-sided*, bottom).

3.7 The eROC curve. `hROC()`, `plot_funregions()` and `plot_regions()` functions

For illustrative purposes, following code and figures study the capacity of improving the classification performance of the gene 20202438 expression intensity via functional transformations and its impact on the final decision rules. Different approaches reviewed in Section 1.2.2 are included. The first one considering an ordinary cubic polynomial formula (`hroc_cubic` object), and a linear tail-restricted cubic splines (`hroc_rcs` object) for the right-hand side of logistic regression model. The second one using two different bandwidths, $h = 1$ and $h = 3$ for `hroc_lkr1` and `hroc_lkr3` object, respectively, to estimate density functions of the marker for both positive and negative populations by using kernel density estimation. The third one, considering the overfitting transformation estimation, $\widehat{h^*_{of}}(\cdot)$ (`hroc_overfit` object).

```
R> hroc_cubic <- hROC(X, D)
R> hroc_cubic

Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A logistic regression model of the form D ~ pol(X,3) has been performed.
The estimated parameters of the model are the following:
Intercept         X       X^2       X^3
 "-204.2"   "999.5" "-1573.7"   "804.0"
The specificity and sensitivity reported by the Youden index are 0.677 and 0.694,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.718) U (0.795, Inf).
The area under the ROC curve (AUC) is 0.725.


R> hroc_rcs <- hROC(X, D, formula.lrm = "D ~ rcs(X,8)")
R> hroc_rcs

Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A logistic regression model of the form D ~ rcs(X,8) has been performed.
The estimated parameters of the model are the following:
   Intercept            X           X'          X''         X'''        X''''
     "9.602"    "-13.934"     "11.761"  "-1809.932"  "10205.202" "-10407.198"
       X'''''                 X''''''
   "-8660.667"             "21802.573"
The specificity and sensitivity reported by the Youden index are 0.677 and 0.742,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.74) U (0.8, Inf).
The area under the ROC curve (AUC) is 0.737.


R> hroc_lkr1 <- hROC(X, D, type = "kernel")
R> hroc_lkr1

Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.694 and 0.71,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.586) U (0.623, 0.745) U (0.802, Inf).
The area under the ROC curve (AUC) is 0.750.


R> hroc_lkr3 <- hROC(X, D, type = "kernel", kernel.h = 3)
R> hroc_lkr3

Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
The specificity and sensitivity reported by the Youden index are 0.661 and 0.742,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.746) U (0.8, Inf).
The area under the ROC curve (AUC) is 0.732.


R> hroc_overfit <- hROC(X, D, type = "overfitting")
R> hroc_overfit

Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
The overfitted ROC curve is reported.
The specificity and sensitivity reported by the Youden index are 1.00 and 1.00,
 respectively, corresponding to the following classification subset:
 (-Inf, 0.54) U (0.639, 0.639) U (0.66, 0.693) U (...) U (0.835, 0.835) U (0.837, Inf).
The area under the ROC curve (AUC) is 1.000.
```

Following chunk of code compares the AUC achieved from each transformation of the gene 20202438 considered:

```
R> list_hroc <- list(Cubic = hroc_cubic, Splines = hroc_rcs,
+                    LikRatioEst_h1 = hroc_lkr1, LikRatioEst_h3 = hroc_lkr3,
+                    gAUC_restC = groc_C, Overfit = hroc_overfit)
R> AUCs <- sapply(list_hroc, function(x) x$auc)
R> round(AUCs, 3)
```

```
  Cubic        Splines LikRatioEst_h1 LikRatioEst_h3     gAUC_restC        Overfit
  0.725          0.737          0.750          0.732          0.762          1.000
```

Note that `type = "overfitting"` computes the transformation providing a perfect classification region with no restrictions and completely based on the available sample. The resulting AUC is 1 if there are no ties between positive and negative values (as this is the case), but the overfitting is clear and the underlying rules may not be extended to any other data.

By using the `plot_funregions()` function as displayed in the code snippet below, Figure 9 represents the different functional transformations estimated previously.

```
R> par(mfrow = c(2,3), mar = c(3.1,2.1,4.1,1.1))
R> for(i in seq_along(list_hroc)){
+   main <- NULL
+   if(i == 5) main <- "General approach \n under restriction (C)"
+   plot_funregions(list_hroc[[i]], FPR = .1,  main = main)
+ }
```

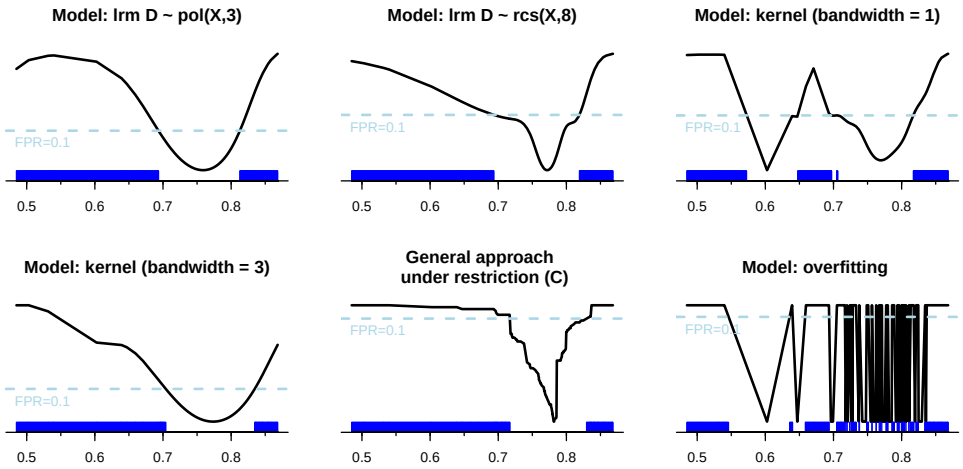


Figure 9: Different functional transformations and resulting classification subsets for gene 20202438. Classification regions for FPR 0.1 are remarked in blue color. Top, from left to right: cubic polynomial function, restricted cubic splines, and likelihood ratio estimation (LRE) with bandwidth 1. Bottom: LRE with bandwidth 3, transformation resulting in gROC curve under restriction *(C)*, and overfittintg.

Finally, using the `plot_regions()` function, Figure 10 shows the resulting classification subsets over the original space for five of the six methods above (overfitting transformation was excluded because the rules are unions of many intervals). Particularly, the decision rules reporting a FPR of 0.1 are highlighted in blue color. All the methods except the third one (likelihood ratio estimation with bandwidth 1) report classification rules of the form $s_t = (-\infty, x_t^L] \cup (x_t^U, \infty)$. The AUCs range from 0.725 to 0.76, compared to 0.547 by the empirical standard ROC curve for the gene under study.

The third method is not very convenient because it does not report a significant gain in AUC, but the shape of some classification rules is more complex: $s_t = (-\infty, a_t] \cup (b_t, c_t] \cup (d_t, \infty)$ or even as an union of 4 intervals (as it may be seen for $t = 0.1$).

```
R> for(i in 1:5){
+   if(i == 1){
+     plot_regions(list_hroc[[i]], FPR = .1, cex.legend = 1.5, plot.auc = TRUE)
+   }else if(i < 5){
+     plot_regions(list_hroc[[i]], FPR = .1, legend = FALSE, plot.auc = TRUE)
```

```
+   }else{
+     plot_regions(list_hroc[[i]], FPR = .1, legend = FALSE, plot.auc = TRUE,
+            main = "Classification subsets: General approach with restriction (C)",
+            main.plotroc = "gROC curve",  xlab = "Gene 20202438 expression intensity")
+   }
+ }
```

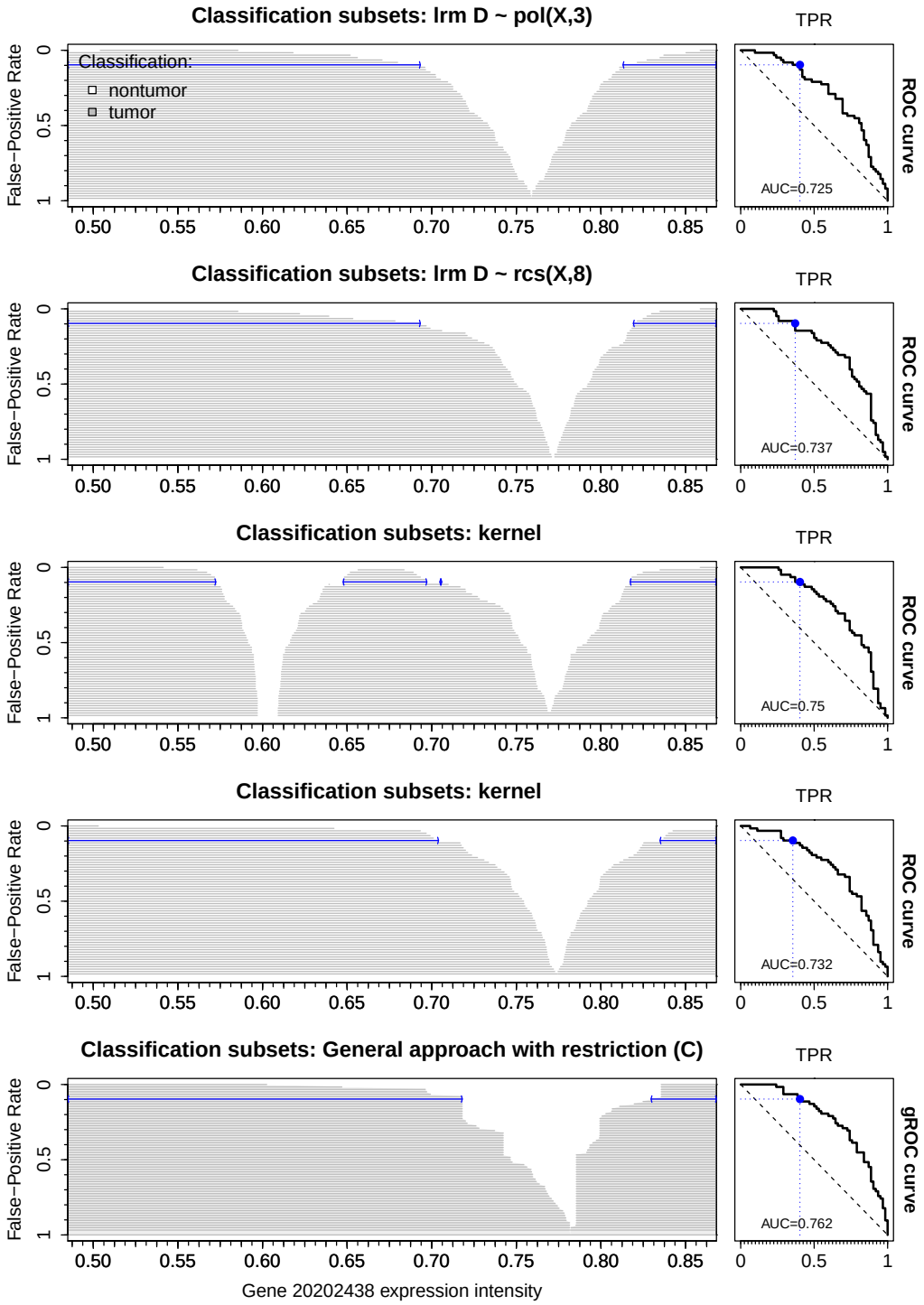


Figure 10: Classification regions and the resulting ROC curve (90º rotated) for the gene 20202438. From top to bottom: i) ROC curve for cubic transformation, ii) ROC curve for restricted cubic splines transformation with 8 knots, iii) ROC curve for likelihood ratio estimation with bandwidth 1, iv) ROC curve for likelihood ratio estimation with bandwidth 3, v) gROC curve under restriction *(C)* for the original marker.

3.8 The multivariate ROC curve

For the next subsections, gene expression intensity of the gene 20202438 will be combined with the gene 18384097 first (Section 3.8.1) and also with the gene 03515901 later on (Section 3.9.1). The goal is to increase the binary classification accuracy and display the resulting decision rules.

3.8.1 A bivariate marker ($p = 2$). `multiROC()` and `plot()` functions

```
R> X <- HCC[ ,c("cg20202438", "cg18384097")]; D <- HCC$tumor
```

Following two estimates consider fitting a binary logistic regression model to combine the two markers. First, by using the formula by default at right-hand side of the regression model; second, considering restricted cubic splines with 4 knots for each univariate marker.

```
R> biroc_lrm <- multiROC(X, D)
R> biroc_lrm
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
A stepwise logistic regression model from the initial formula D ~ X.1 + I(X.1^2) + X.2
 + I(X.2^2) + I(X.1*X.2) has been performed.
The estimated parameters of the resulting model are the following:
(Intercept)         X.1    I(X.1^2)         X.2    I(X.2^2)
   78.87003  -158.65880   106.44235   -69.18716    56.79345
The specificity and sensitivity reported by the Youden index are 0.871 and 0.645,
 respectively, corresponding to the cut-off point -0.0291 for the transformation h(X)
 in the formula above.
The area under the ROC curve (AUC) is 0.774.
```

```
R> biroc_lrm_rcs <- multiROC(X, D, method = "lrm",
+                formula.lrm = "D ~ rcs(X.1,4) + rcs(X.2,4)")
R> biroc_lrm_rcs
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
A stepwise logistic regression model from the initial formula
 D ~ rcs(X.1,4) + rcs(X.2,4) has been performed.
The estimated parameters of the resulting model are the following:
     (Intercept)   rcs(X.1, 4)X.1  rcs(X.1, 4)X.1' rcs(X.1, 4)X.1''
        6.269163        -6.572278       -15.444407       484.022536
  rcs(X.2, 4)X.2  rcs(X.2, 4)X.2' rcs(X.2, 4)X.2''
       -2.394454        -7.067841       252.533859
The specificity and sensitivity reported by the Youden index are 0.919 and 0.629,
 respectively, corresponding to the cut-off point 0.262 for the transformation h(X)
 in the formula above.
The area under the ROC curve (AUC) is 0.844.
```

The next two estimates consider a linear combination of the two markers with fixed parameters. Those coefficients are estimated by Pepe and Thompson (2000) method (first) and by Su and Liu (1993) approach (second).

```
R> biroc_PT <- multiROC(X, D, method = "fixedLinear", methodLinear = "PepeThompson")
R> biroc_PT
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
A linear combination with fixed parameters estimated by PepeThompson approach has been
 considered.
The specificity and sensitivity reported by the Youden index are 0.871 and 0.548,
 respectively, corresponding to the cut-off point 0.358 for the transformation
 h(X) =  - 0.51*cg20202438 + 1*cg18384097.
The area under the ROC curve (AUC) is 0.697.
```

```
R> biroc_SL <- multiROC(X, D, method = "fixedLinear", methodLinear = "SuLiu")
R> biroc_SL
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
A linear combination with fixed parameters estimated by SuLiu approach has been
 considered.
The specificity and sensitivity reported by the Youden index are 0.887 and 0.565,
 respectively, corresponding to the cut-off point 1 for the transformation
 h(X) =  - 0.547*cg20202438 + 1.91*cg18384097.
The area under the ROC curve (AUC) is 0.694.
```

The following object keeps the ROC curve and decision rules when linear combinations of the two markers under study are considered, but with dynamic parameters (each $t \in [0,1]$ may result in a different linear combination) by using the Meisner et al. (2021) algorithm.

```
R> biroc_Meis <- multiROC(X, D, method = "dynamicMeisner")
R> biroc_Meis
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
A linear combination with dynamic parameters has been considered.
The specificity and sensitivity reported by the Youden index are 0.968 and 0.50,
 respectively, corresponding to the cut-off point 0.701 for the transformation
 h(X) =  - 0.0884*cg20202438 + 0.996*cg18384097.
The area under the ROC curve (AUC) is 0.702.
```

Finally, the optimal transformation is estimated based on bivariate kernel density estimation by Martínez-Camblor et al. (2021b), which uses the `kde()` function in the ks package with the `"Hbcv"` method by default (input `kernelOptimal.H`) to approximate the bandwidth matrix of the bivariate marker in each group.

```
R> biroc_kernel <- multiROC(X, D, method = "kernelOptimal")
R> biroc_kernel
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 2 variables have been considered.
Optimal transformation based on bivariate kernel density estimation has been considered.
Hbcv method has been used to estimate the bandwidth matrix in each group.
The specificity and sensitivity reported by the Youden index are 0.968 and 0.677,
 respectively, corresponding to the cut-off point 0.481 for the transformation computed
 in optimalT(X).
The area under the ROC curve (AUC) is 0.872.
```

```
R> ls(biroc_kernel)
```

```
 [1] "auc"      "c"        "cases"    "controls" "D"        "H.method" "levels"
 [8] "method"   "optimalT" "roc"      "t"        "X"        "Z"
```

```
R> class(biroc_kernel$optimalT)
```

```
[1] "function"
```

A quick comparison of the empirical reported AUCs shows that the best result is the kernel density estimation of the bivariate marker and the logistic regression model considering restricted cubic splines. Both are complex transformations and we will see that result in complex decision rules which are difficult to interpret (Figure 12 (c)-(d)).

```
R> list_biroc <- list(LRm = biroc_lrm, LRm_rcs = biroc_lrm_rcs,
+                     PepeTh = biroc_PT, SuLiu = biroc_SL,
+                     Meisner = biroc_Meis, KernelDens = biroc_kernel)
R> AUCs <- sapply(list_biroc, function(x) x$auc)
R> round(AUCs, 3)
```

```
       LRm    LRm_rcs     PepeTh      SuLiu    Meisner KernelDens
     0.774      0.844      0.697      0.694      0.702      0.872
```

The resulting empirical ROC curves can be graphically displayed (Figure 11) with the `plot()` function of our package:

```
R> colors <- c(LRm = "#4357AD", LRm_rcs = "#48A9A6", PepeTh = "gold",
+              SuLiu = "#E18D10", Meisner = "#C96066", KernelDens = "violet")

R> plot(biroc_PT, main = "Bivariate ROC curve\n(linear transf.)", lwd = 3,
+      col = colors["PepeTh"])
> plot(biroc_SL, new = FALSE, lwd = 3, col = colors["SuLiu"])
> plot(biroc_Meis, new = FALSE, lwd = 3, col = colors["Meisner"])
> legend("bottomright", c("Pepe&Thompson", "Su&Liu", "Meisner et al."),
+        col = colors[c("PepeTh","SuLiu","Meisner")], lwd = 3, bty = "n", inset = .01)

R> plot(biroc_lrm, main = "Bivariate ROC curve\n(complex transf.)", lwd = 3,
+      col = colors["LRm"])
> plot(biroc_lrm_rcs, new = FALSE, lwd = 3, col = colors["LRm_rcs"])
> plot(biroc_kernel, new = FALSE, lwd = 3, col = colors["KernelDens"])
> legend("bottomright", c("LogReg", "LogReg RCS", "Kernel optimal"),
+        col = colors[c("LRm","LRm_rcs","KernelDens")], lwd = 3, bty = "n", inset = .01)
```

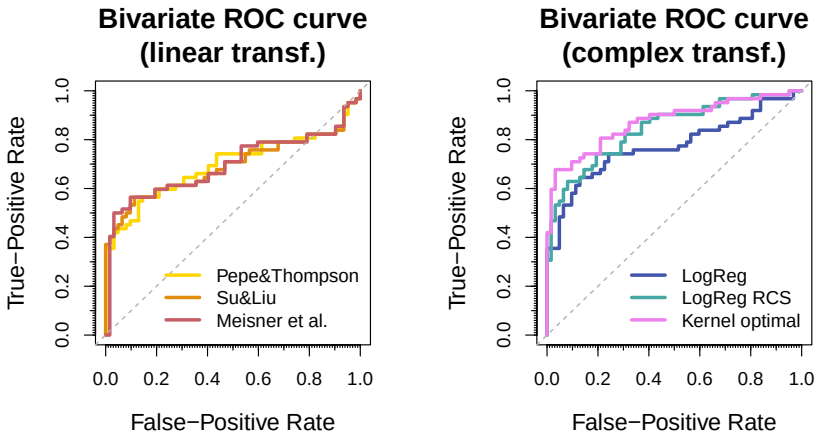


Figure 11: Empirical ROC curves for the combination of the genes 20202438 and 18384097. By using linear combinations (left), and more complex transformations (right).

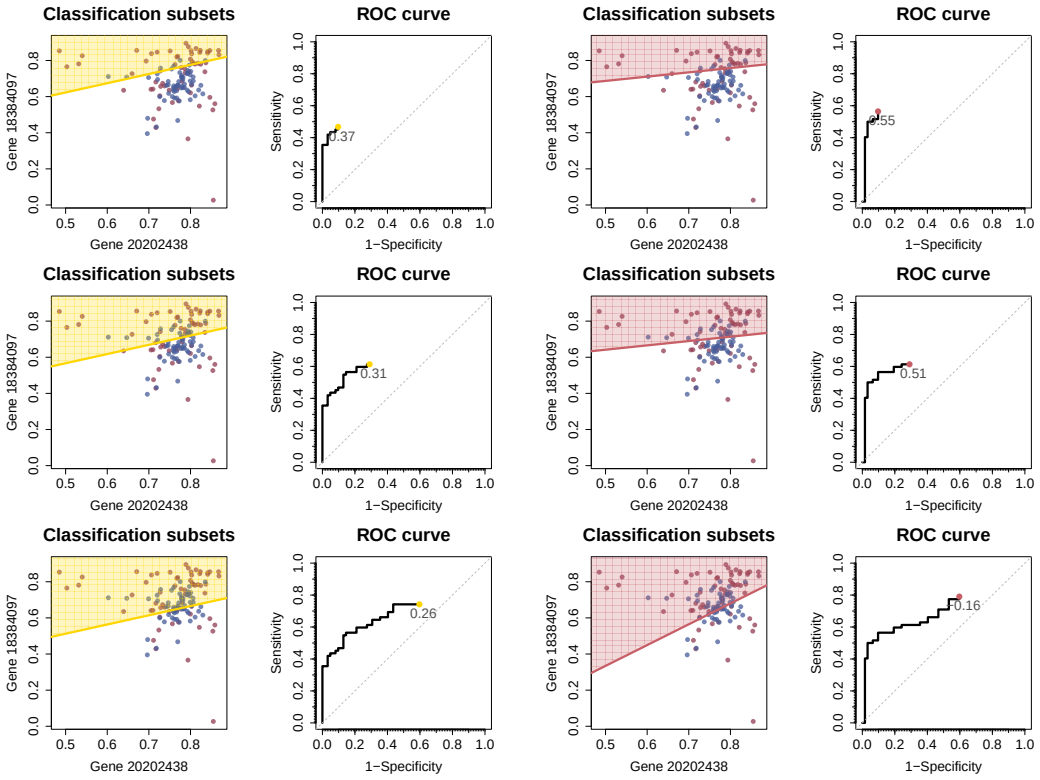
3.9 Videos for the bivariate ROC curve. `movieROC()` and `plot_buildROC()` functions

The following line of code saves video as GIFs with the construction of the empirical ROC curve for the bivariate marker (gene 20202438, gene 18384097) by using different methods to combine the two univariate markers. One video is saved for each method in the object `list_biroc` created in the previous section.

```
R> for(x in names(list_biroc)){
+   movieROC(list_biroc[[x]], display.method = "OV",
+            xlab = "Gene 20202438", ylab = "Gene 18384097",
+            border = TRUE, lwd.curve = 4, cex = 1.2, col.threshold = colors[x],
+            file = paste0("Video_", x, ".gif"))
+ }
```

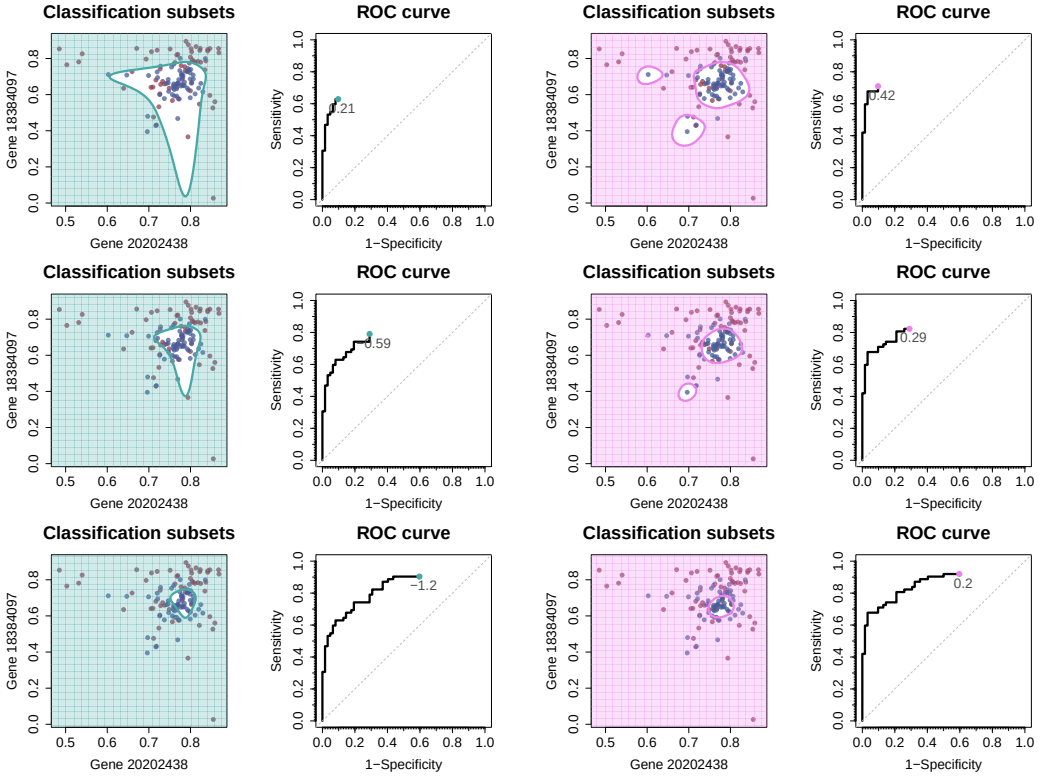
The three snapshots for each method illustrated in Figure 12 can also be generated by the `plot_buildROC()` function, as follows:

```
R> for(x in names(list_biroc)){
+   for(fpr in c(.1, .3, .6)){
+     plot_buildROC(list_biroc[[x]], display.method = "OV", FPR = fpr,
+                   build.process = TRUE, completeROC = FALSE,
+                   xlab = "Gene 20202438", ylab = "Gene 18384097",
+                   border = TRUE, lwd.curve = 4, cex = 1.2, col.threshold = colors[x])
+   }
+ }
```



(a) Linear combinations with fixed parameters by Pepe and Thompson (2000).

(b) Linear combinations with dynamic parameters by Meisner et al. (2021).



(c) Logistic regression model with restricted cubic splines with 4 knots for each univariate marker.

(d) Optimal transformation by multivariate kernel density estimation with "Hbcv" method by default (Martínez-Camblor et al., 2021b).

Figure 12: Snapshots (from `movieROC()` function) of the classification procedure and empirical ROC curve for the bivariate marker (gene 20202438, gene 18384097) when FPR $\in \{0.1, 0.3, 0.6\}$.

3.9.1 A multidimensional marker with $p > 2$

For this subsection, a marker of dimension $p = 3$ is considered. Particularly, gene expression intensity of the gene 20202438 is combined with the gene 18384097 and the gene 03515901. The methods used are those studied in the previous subsection for $p = 2$.

```
R> X <- HCC[ ,c("cg20202438", "cg18384097", "cg03515901")]; D <- HCC$tumor
```

Following two estimates consider fitting a binary logistic regression model to combine the three markers. First, by using the formula by default extended to dimension 3 at right-hand side of the regression model; second, considering restricted cubic splines with 4 knots for each univariate marker.

```
R> multiroc_lrm <- multiROC(X, D, method = "lrm", formula.lrm =
+             "D ~ X.1 + I(X.1^2) + X.2 + I(X.2^2) + X.3 + I(X.3^2) + I(X.1*X.2*X.3)")
R> multiroc_lrm
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
A stepwise logistic regression model from the initial formula
 D ~ X.1 + I(X.1^2) + X.2 + I(X.2^2) + X.3 + I(X.3^2) + I(X.1*X.2*X.3)
 has been performed.
The estimated parameters of the resulting model are the following:
(Intercept)         X.1    I(X.1^2)         X.2    I(X.2^2)         X.3    I(X.3^2)
   98.57338  -185.76142   127.89683   -41.25060    32.14067   -51.56556    33.29841
The specificity and sensitivity reported by the Youden index are 0.887 and 0.806,
 respectively, corresponding to the cut-off point -0.301 for the transformation h(X)
 in the formula above.
The area under the ROC curve (AUC) is 0.876.
```

```
R> multiroc_lrm_rcs <- multiROC(X, D, method = "lrm",
+                               formula.lrm = "D ~ rcs(X.1,4) + rcs(X.2,4) + rcs(X.3,4)")
R> multiroc_lrm_rcs
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
A stepwise logistic regression model from the initial formula
 D ~ rcs(X.1,4) + rcs(X.2,4) + rcs(X.3,4) has been performed.
The estimated parameters of the resulting model are the following:
     (Intercept)   rcs(X.2, 4)X.2  rcs(X.2, 4)X.2' rcs(X.2, 4)X.2''   rcs(X.3, 4)X.3
    14.696335839     -0.003217415    -21.304677940    336.242585415    -27.944341462
 rcs(X.3, 4)X.3' rcs(X.3, 4)X.3''
    18.022003883    -50.586827581
The specificity and sensitivity reported by the Youden index are 0.871 and 0.855,
 respectively, corresponding to the cut-off point -0.607 for the transformation h(X)
 in the formula above.
The area under the ROC curve (AUC) is 0.888.
```

The next two estimates consider a linear combination of the three markers with fixed parameters. Those coefficients are estimated by Pepe and Thompson (2000) method (first) and by Su and Liu (1993) approach (second).

```
R> multiroc_PT <- multiROC(X, D, method = "fixedLinear", methodLinear = "PepeThompson")
R> multiroc_PT
```

```
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
A linear combination with fixed parameters estimated by PepeThompson approach has been
 considered.
The specificity and sensitivity reported by the Youden index are 0.855 and 0.742,
 respectively, corresponding to the cut-off point -0.0755 for the transformation
 h(X) =  0.81*cg20202438 - 0.1*cg18384097 - 1*cg03515901.
The area under the ROC curve (AUC) is 0.811.
```

```
R> multiroc_SL <- multiROC(X, D, method = "fixedLinear", methodLinear = "SuLiu")
R> multiroc_SL
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
A linear combination with fixed parameters estimated by SuLiu approach has been
 considered.
The specificity and sensitivity reported by the Youden index are 0.968 and 0.597,
 respectively, corresponding to the cut-off point -0.408 for the transformation
 h(X) =  1.96*cg20202438 + 0.594*cg18384097 - 4.22*cg03515901.
The area under the ROC curve (AUC) is 0.797.
```

The following object keeps the ROC curve and decision rules when linear combinations of the three markers under study are considered, but with dynamic parameters (each $t \in [0, 1]$ may result in a different linear combination) by using the Meisner et al. (2021) algorithm.

```
R> multiroc_Meis <- multiROC(X, D, method = "dynamicMeisner")
R> multiroc_Meis
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
A linear combination with dynamic parameters has been considered.
The specificity and sensitivity reported by the Youden index are 0.855 and 0.774,
 respectively, corresponding to the cut-off point -0.407 for the transformation
 h(X) =  0.49*cg20202438 - 0.438*cg18384097 - 0.754*cg03515901.
The area under the ROC curve (AUC) is 0.824.
```

Finally, the optimal transformation is estimated based on multivariate kernel density estimation by Martínez-Camblor et al. (2021b), which uses the `kde()` function in the **ks** package with the `"Hpi"` method by default (input `kernelOptimal.H`) to approximate the bandwidth matrix of the multivariate marker in each group.

```
R> multiroc_kernel <- multiROC(X, D, method = "kernelOptimal")
R> multiroc_kernel
Data was encoded with nontumor (controls) and tumor (cases).
There are 62 controls and 62 cases.
A total of 3 variables have been considered.
Optimal transformation based on bivariate kernel density estimation has been considered.
Hpi method has been used to estimate the bandwidth matrix in each group.
The specificity and sensitivity reported by the Youden index are 0.984 and 0.968,
 respectively, corresponding to the cut-off point 0.189 for the transformation
 computed in optimalT(X).
The area under the ROC curve (AUC) is 0.986.
```

A quick comparison of the empirical reported AUCs shows that the best result is the kernel density estimation of the bivariate marker and the logistic regression model considering restricted cubic splines. Both are complex transformations and we will see that result in complex decision rules which are difficult to interpret (Figure 16).

```
R> list_multiroc <- list(LRm = multiroc_lrm, LRm_rcs = multiroc_lrm_rcs,
+                    PepeTh = multiroc_PT, SuLiu = multiroc_SL,
+                    Meisner = multiroc_Meis, KernelDens = multiroc_kernel)
R> AUCs <- sapply(list_multiroc, function(x) x$auc)
R> round(AUCs, 3)
       LRm    LRm_rcs     PepeTh      SuLiu    Meisner KernelDens
     0.876      0.888      0.811      0.797      0.824      0.986
```

The resulting empirical ROC curves can be graphically displayed (Figure 13) with the `plot()` function of our package:

```
R> plot(multiroc_PT, main = "Multivariate ROC curve\n(linear transf.)", lwd = 3,
+      col = colors["PepeTh"])
> plot(multiroc_SL, new = FALSE, lwd = 3, col = colors["SuLiu"])
> plot(multiroc_Meis, new = FALSE, lwd = 3, col = colors["Meisner"])
> legend("bottomright", c("Pepe&Thompson", "Su&Liu", "Meisner et al."),
+        col = colors[c("PepeTh","SuLiu","Meisner")], lwd = 3, bty = "n", inset = .01)
```

```
R> plot(multiroc_lrm, main = "Multivariate ROC curve\n(complex transf.)", lwd = 3,
+      col = colors["LRm"])
> plot(multiroc_lrm_rcs, new = FALSE, lwd = 3, col = colors["LRm_rcs"])
> plot(multiroc_kernel, new = FALSE, lwd = 3, col = colors["KernelDens"])
> legend("bottomright", c("LogReg", "LogReg RCS", "Kernel optimal"),
+        col = colors[c("LRm","LRm_rcs","KernelDens")], lwd = 3, bty = "n", inset = .01)
```

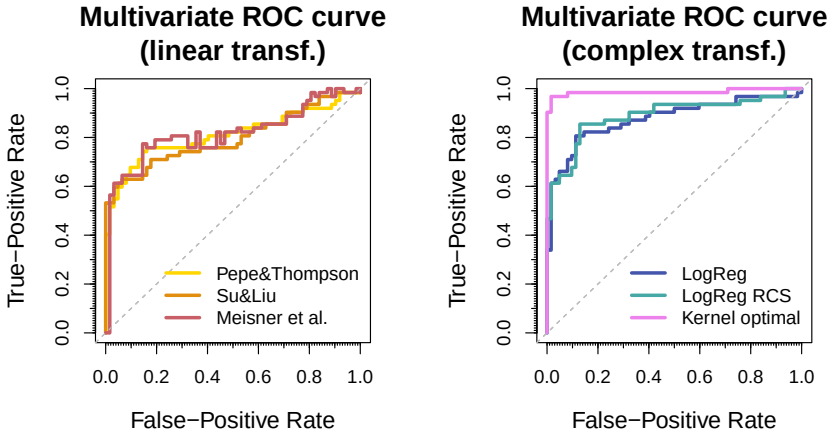


Figure 13: Empirical ROC curves for the combination of the genes 20202438, 18384097 and 03515901. By using linear combinations (left), and more complex transformations (right).

Thanks to Figure 13 one can clearly see that the ROC curve estimate proposed by Meisner et al. (2021) (using their R function `maxTPR()`) may not be monotone.

3.10 Videos for the multivariate ROC curve. `movieROC()` and `plot_buildROC()` functions

The following line of code saves video as GIFs with the construction of the empirical ROC curve for the multivariate marker (gene 20202438, gene 18384097, gene 03515901) by using different methods to combine the three univariate markers. One video is saved for each method in the object `list_multiroc` created in the previous section.

When the marker has a dimension higher than two it is difficult to visualize the data and the classification regions. Therefore, the `plot_buildROC()` and `movieROC()` functions offer two options for showing the results, both on a bidimensional space. On the one hand, to choose two of the components of the multivariate marker and project the classification subsets on the plain defined by them (Figure 14, first `for` loop in the code below). On the other, to project the classification regions on the plain defined by the two first principal components (Figure 15, second `for` loop in the code below).

```
R> x <- "PepeTh"

R> for(fpr in c(.1, .3, .6)){
+    plot_buildROC(list_multiroc[[x]], display.method = "OV", displayOV = c(1,2),
+                  FPR = fpr, build.process = TRUE, completeROC = FALSE, border = TRUE,
+                  lwd.curve = 4, cex = 1.2, col.threshold = colors[x])
+    plot_buildROC(list_multiroc[[x]], display.method = "OV", displayOV = c(1,3),
+                  FPR = fpr, build.process = TRUE, completeROC = FALSE, border = TRUE,
+                  lwd.curve = 4, cex = 1.2, col.threshold = colors[x])
+    plot_buildROC(list_multiroc[[x]], display.method = "OV", displayOV = c(2,3),
+                  FPR = fpr, build.process = TRUE, completeROC = FALSE, border = TRUE,
+                  lwd.curve = 4, cex = 1.2, col.threshold = colors[x])
+ }

R> for(fpr in c(.1, .3, .6, .9)){
+    plot_buildROC(list_multiroc[[x]], display.method = "PCA",
+                  FPR = fpr, build.process = TRUE, completeROC = FALSE, border = TRUE,
+                  lwd.curve = 4, cex = 1.2, col.threshold = colors[x])
+ }
```

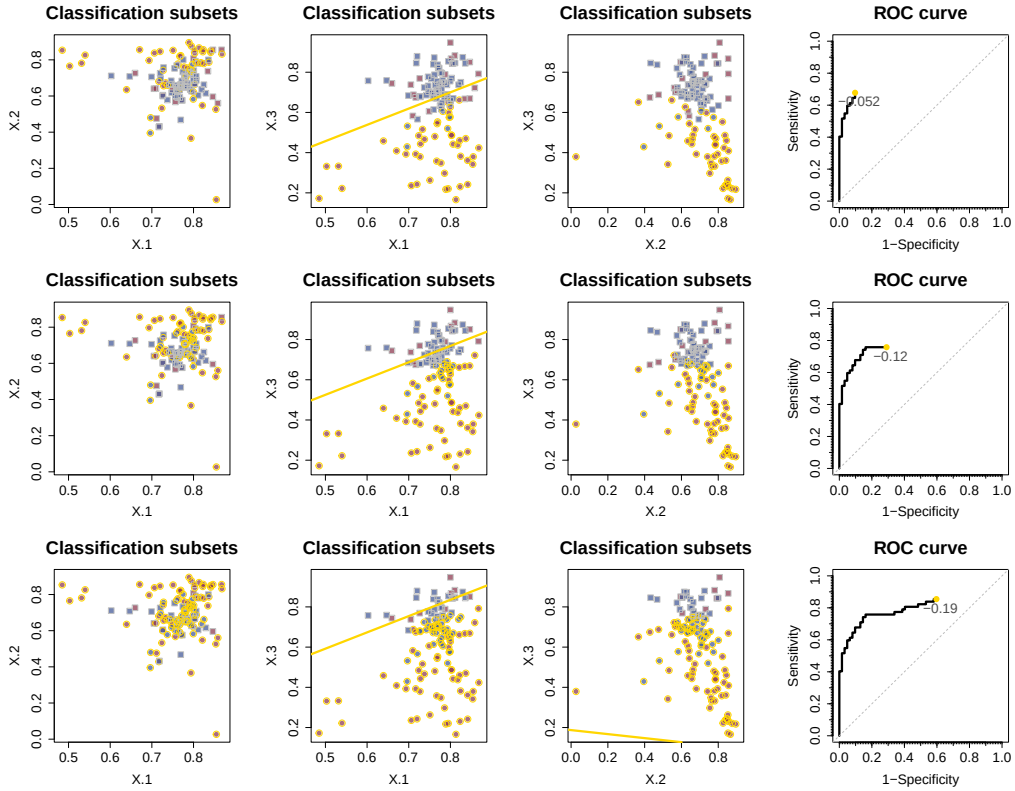


Figure 14: Multivariate ROC curve estimation for the simultaneous diagnostic accuracy of genes 20202438, 18384097 and 03515901. Pepe and Thompson (2000) approach was used and classification rules when FPR $\in \{0.1, 0.3, 0.6\}$ (from top to bottom). The classification subsets are projected over the three pairs of original univariate markers (1-2, 1-3, 2-3, from left to right); points in gold color for positive classification and in gray for negative.

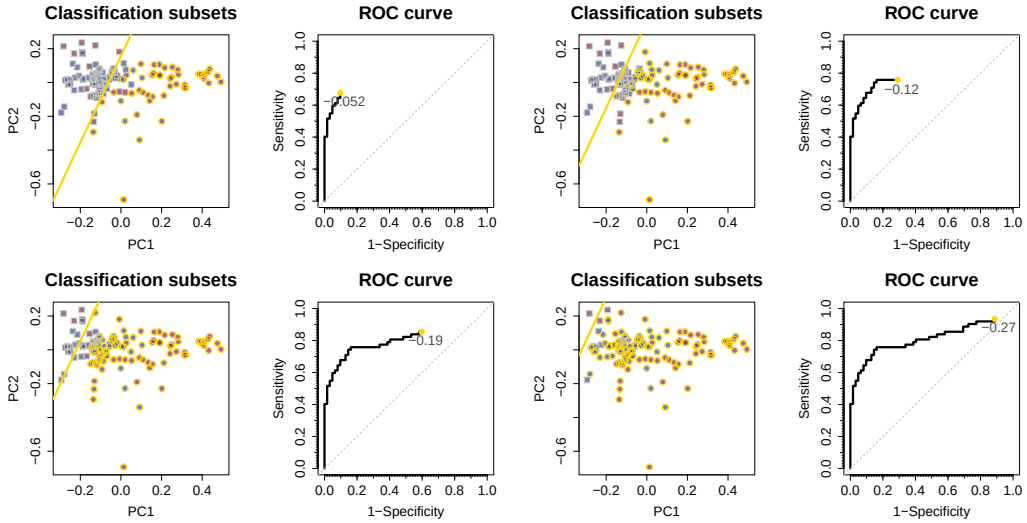


Figure 15: Multivariate ROC curve estimation for the simultaneous diagnostic accuracy of genes 20202438, 18384097 and 03515901. Pepe and Thompson (2000) approach was used and classification rules when FPR $\in \{0.1, 0.3, 0.6, 0.9\}$. The classification subsets are projected over the two first principal components.

With comparison purposes, Figure 14 is repeated for the logistic regression model considering restricted cubic splines, with the R code showed in the first `for` loop above with `x <-"LRm_rcs"`. The frontiers defining the decision rules are clearly more complex in this case.

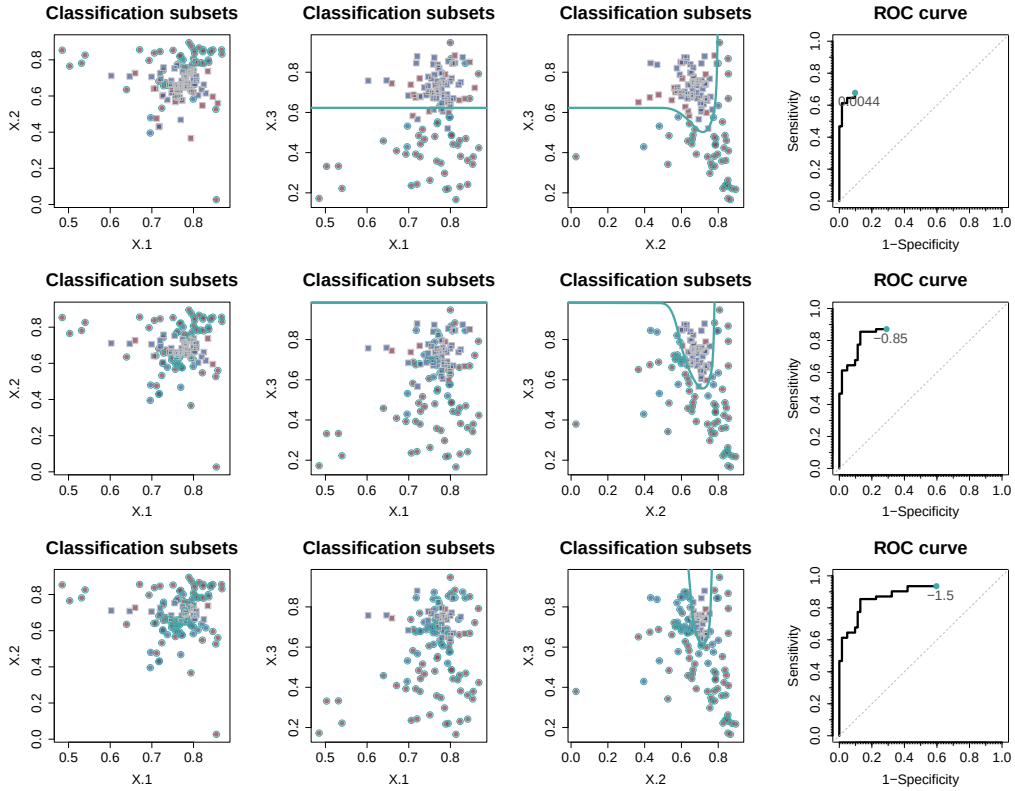


Figure 16: Multivariate ROC curve estimation for the simultaneous diagnostic accuracy of genes 20202438, 18384097 and 03515901. A logistic regression model considering restricted cubic splines was used and classification rules when FPR $\in \{0.1, 0.3, 0.6\}$ (from top to bottom). The classification subsets are projected over the three pairs of original univariate markers (1-2, 1-3, 2-3, from left to right); points in blue color for positive classification and in gray for negative.

Finally, the next code snippet saves a video with the construction of the multivariate ROC curve illustrated in the third column of Figure 16, i.e. with the classification subsets projected over the genes 18384097 and 03515901. The resulting GIF file is called `video_multiROC_LRm_rcs_proj23.gif`.

```
R> x <- "LRm_rcs"
R> movieROC(list_multiroc[[x]], display.method = "OV", displayOV = c(2,3),
+          border = TRUE, lwd.curve = 4, cex = 1.2, col.threshold = colors[x],
+          file = "video_multiROC_LRm_rcs_proj23.gif")
```

Bibliography

D. Dorfman, K. Berbaum, C. Metz, R. Lenth, J. Hanley, and H. Dagga. Proper receiver operating characteristic analysis: The bigamma model. *Academic Radiology*, 4(2):138–149, 1997. URL https://doi.org/10.1016/s1076-6332(97)80013-x. [p2]

T. Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R. *Journal of Statistical Software*, 21(7):1–16, 2007. URL https://doi.org/10.18637/jss.v021.i07. [p5, 8]

R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5:345–345, 1962. URL https://doi.org/10.1145/367766.368168. [p3, 7]

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982. URL https://doi.org/110.1148/radiology.143.1.7063747. [p2]

F. E. Harrell Jr. *rms: Regression Modeling Strategies*, 2023. URL https://CRAN.R-project.org/package=rms. R package version 6.7-1. [p8]

F. Hsieh and B. W. Turnbull. Nonparametric and semiparametric estimation of the receiver operating characteristic curve. *The Annals of Statistics*, 24(1):25–40, 1996. URL https://doi.org/10.1214/aos/1033066197. [p2]

H. Kauppi. The generalized receiver operating characteristic curve. Discussion paper 114, Aboa Centre for Economics, 2016. URL https://www.econstor.eu/bitstream/10419/233329/1/aboa-ce-dp114.pdf. [p4]

C. Liu, A. Liu, and S. Halabi. A min–max combination of biomarkers to improve diagnostic accuracy. *Statistics in Medicine*, 30(16):2005–2014, 2011. URL https://doi.org/10.1002/sim.4238. [p7]

P. Martínez-Camblor and J. C. Pardo-Fernández. Parametric estimates for the receiver operating characteristic curve generalization for non-monotone relationships. *Statistical Methods in Medical Research*, 28(7):2032–2048, 2019. URL https://doi.org/10.1177/0962280217747009. [p4]

P. Martínez-Camblor, N. Corral, C. Rey, J. Pascual, and E. Cernuda-Morollón. Receiver operating characteristic curve generalization for non-monotone relationships. *Statistical Methods in Medical Research*, 26(1):113–123, 2017. URL https://doi.org/10.1177/0962280214541095. [p2]

P. Martínez-Camblor, S. Pérez-Fernández, and S. Díaz-Coto. Improving the biomarker diagnostic capacity via functional transformations. *Journal of Applied Statistics*, 46(9):1550–1566, 2019. URL https://doi.org/10.1080/02664763.2018.1554628. [p4, 5]

P. Martínez-Camblor, S. Pérez-Fernández, and S. Díaz-Coto. The area under the generalized receiver-operating characteristic curve. *The International Journal of Biostatistics*, 18(1):293–306, 2021a. URL https://doi.org/10.1515/ijb-2020-0091. [p5]

P. Martínez-Camblor, S. Pérez-Fernández, and S. Díaz-Coto. Optimal classification scores based on multivariate marker transformations. *AStA Advances in Statistical Analysis*, 105(4):581–599, 2021b. URL https://doi.org/10.1007/s10182-020-00388-z. [p4, 5, 21, 23, 25]

M. W. McIntosh and M. S. Pepe. Combining several screening tests: Optimality of the risk score. *Biometrics*, 58(3):657–664, 2002. URL https://doi.org/10.1111/j.0006-341X.2002.00657.x. [p4]

A. Meisner, M. Carone, M. S. Pepe, and K. F. Kerr. Combining biomarkers by maximizing the true positive rate for a fixed false positive rate. *Biometrical Journal*, 63(6):1223–1240, 2021. URL https://doi.org/10.1002/bimj.202000210. [p7, 21, 23, 25, 26]

D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2023. URL https://CRAN.R-project.org/package=e1071. R package version 1.7-13. [p3]

M. S. Pepe and M. L. Thompson. Combining diagnostic test results to increase accuracy. *Biostatistics*, 1 (2):123–140, 2000. URL https://doi.org/10.1093/biostatistics/1.2.123. [p7, 20, 23, 24, 27]

S. Pérez-Fernández, P. Martínez-Camblor, P. Filzmoser, and N. Corral. Visualizing the decision rules behind the ROC curves: understanding the classification process. *AStA Advances in Statistical Analysis*, 105(1):135–161, 2021. URL https://doi.org/10.1007/s10182-020-00385-2. [p3, 5]

J. Shen, S. Wang, Y.-J. Zhang, M. Kappil, H.-C. Wu, M. G. Kibriya, Q. Wang, F. Jasmine, H. Ahsan, P.-H. Lee, et al. Genome-wide dna methylation profiles in hepatocellular carcinoma. *Hepatology*, 55(6): 1799–1808, 2012. URL https://doi.org/10.1002/hep.25569. [p9]

J. Q. Su and J. S. Liu. Linear combinations of multiple diagnostic markers. *Journal of the American Statistical Association*, 88(424):1350–1355, 1993. URL https://doi.org/10.1080/01621459.1993.10476417. [p7, 20, 24]

W. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950. URL https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3. [p2]

K. H. Zou, W. J. Hall, and D. E. Shapiro. Smooth non-parametric receiver operating characteristic (ROC) curves for continuous diagnostic tests. *Statistics in Medicine*, 16(19):2143–2156, 1997. URL https://doi.org/10.1002/(SICI)1097-0258(19971015)16:19<2143::AID-SIM655>3.0.CO;2-3. [p2]