

# Package ‘movieROC’

June 28, 2024

**Type** Package

**Title** Visualizing the Decision Rules Underlying Binary Classification

**Version** 0.1.1

**Description** Visualization of decision rules for binary classification and Receiver Operating Characteristic (ROC) curve estimation under different generalizations proposed in the literature:

- making the classification subsets flexible to cover those scenarios where both extremes of the marker are associated with a higher risk of being positive, considering two thresholds (gROC() function);
  - transforming the marker by a proper function trying to improve the classification performance (hROC() function);
  - when dealing with multivariate markers, considering a proper transformation to univariate space trying to maximize the resulting AUC of the TPR for each FPR (multiROC() function).
- The classification regions behind each point of the ROC curve are displayed in both static graphics (plot\_buildROC(), plot\_regions() or plot\_funregions() function) or videos (movieROC() function).

**License** GPL-3

**Encoding** UTF-8

**BugReports** <https://github.com/perezsonia/movieROC/issues>

**Imports** rms, animation, intrval, gtools, e1071, robustbase, Rsolnp, ks, zoo

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5.0)

**NeedsCompilation** no

**Suggests** R.rsp

**VignetteBuilder** R.rsp

**Author** Sonia Perez-Fernandez [aut, cre]  
(<<https://orcid.org/0000-0002-2767-6399>>)

**Maintainer** Sonia Perez-Fernandez <perezsonia@uniovi.es>

**Repository** CRAN

**Date/Publication** 2024-06-28 07:40:02 UTC

## Contents

gROC	2
gROC_param	4
HCC	5
hROC	6
movieROC	8
movieROC2_densities	11
multiROC	13
plot	16
plot_buildROC	18
plot_densities	21
plot_densityROC	22
plot_funregions	24
plot_regions	25
predict	26
print	27
<b>Index</b>	<b>29</b>

---

gROC

*Build a ROC curve for a univariate marker*

---

### Description

This is one of the main functions of the movieROC package. It builds a univariate ROC curve (standard or general) and returns a ‘groc’ object, a list of class ‘groc’. This object can be [printed](#), [plotted](#), or [predicted](#) for a particular point. It may be also passed to other functions: [plot\\_regions\(\)](#), [plot\\_buildROC\(\)](#), [movieROC\(\)](#), [plot\\_densities\(\)](#), and [plot\\_densityROC\(\)](#).

### Usage

```
gROC(X, D, ...)
## Default S3 method:
gROC(X, D, side = c("right", "left", "both", "both2"),
     restric = FALSE, optim = TRUE, t0 = NULL, t0max = FALSE, verbose = FALSE, ...)
```

### Arguments

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".
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), implemented in the allShortestPaths() function in the <b>e1071</b> package. Default: TRUE.

t0	An integer number between 1 and $m + 1$ (where $m$ is the negative sample size). If <code>restric = TRUE</code> , 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.
verbose	If TRUE, a progress bar is displayed for computationally intensive methods. Default: FALSE.
...	Other parameters to be passed. Not used.

### Details

This function's main job is to estimate an ROC curve for a univariate marker under one of these considerations: larger values of the marker are associated with a higher probability of being positive (resulting in the *right-sided* ROC curve,  $\mathcal{R}_r(\cdot)$ ), the opposite (*left-sided* ROC curve,  $\mathcal{R}_l(\cdot)$ ), when both smaller and larger values of the marker are associated with having more probability of being positive (*gROC curve*,  $\mathcal{R}_g(\cdot)$ ), the opposite (*opposite gROC curve*,  $\mathcal{R}_{g'}(\cdot)$ ).

### Value

A list of class 'groc' with the following fields:

controls, cases	Marker values of negative and positive subjects, respectively.
levels	Levels of response values.
side	Type of ROC curve.
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 <code>side = "right"   "left"</code> .
x1, xu	Vectors of marker thresholds resulting in (t, roc) if <code>side = "both"   "both2"</code> .
auc	Area under the curve estimate.
aucfree	Area under the curve estimate without restrictions.
aucs	Area under the curve with restriction ( $C$ ) departing from every false-positive rate, $FPR \in \{0, 1/m, \dots, 1\}$ .

### Dependencies

If `side = "both"` and `optim = TRUE`, the `allShortesPaths()` function in the **e1071** package is used. Also the `combinations()` function in **gtools** and `%[]%` in **intrval**.

### References

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

S. Pérez-Fernández, P. Martínez-Cambor, P. Filzmoser, and N. Corral (2021). “Visualizing the decision rules behind the ROC curves: understanding the classification process”. *AStA Advances in Statistical Analysis*, **105**(1):135-161. DOI: [doi:10.1007/s10182020003852](https://doi.org/10.1007/s10182020003852).

R. W. Floyd (1962) “Algorithm 97: Shortest path”. *Communications of the ACM*, **5**: 345–345. DOI: [doi:10.1145/367766.368168](https://doi.org/10.1145/367766.368168).

## Examples

```
data(HCC)

# ROC curve estimates for gene 03515901 and response tumor
gROC(X = HCC["cg03515901"], D = HCC$tumor) # Standard right-sided ROC curve
gROC(X = HCC["cg03515901"], D = HCC$tumor, side = "left") # Left-sided ROC curve
gROC(X = HCC["cg03515901"], D = HCC$tumor, side = "both") # gROC curve without restrictions

### Warning: Next line of code is time consuming. gROC curve with restriction (C)
gROC(X = HCC["cg03515901"], D = HCC$tumor, side = "both", restric = TRUE)
```

---

gROC\_param

*Build a binormal ROC curve for a univariate marker*

---

## Description

This function builds a univariate ROC curve (standard or general) assuming the binormal scenario with parameters being the sample estimates. It returns a ‘groc’ object, a list of class ‘groc’.

## Usage

```
gROC_param(X, D, side = c("right", "left", "both", "both2"), N = NULL, ...)
```

## Arguments

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	Number indicating the length of the vector of FPR considered to build the ROC curve: $t \in \{0, 1/N, 2/N, \dots, 1\}$ . Default: 1000.
...	Other parameters to be passed. Not used.

## Details

This function’s main job is to estimate an ROC curve for a univariate marker under one of these considerations: larger values of the marker are associated with a higher probability of being positive (resulting in the *right-sided* ROC curve,  $\mathcal{R}_r(\cdot)$ ), the opposite (*left-sided* ROC curve,  $\mathcal{R}_l(\cdot)$ ), when both smaller and larger values of the marker are associated with having more probability of being positive (*gROC curve*,  $\mathcal{R}_g(\cdot)$ ), the opposite (*opposite gROC curve*,  $\mathcal{R}_{g'}(\cdot)$ ).

**Value**

A list of class 'groc' with the following fields:

controls, cases	Marker values of negative and positive subjects, respectively.
levels	Levels of response values.
side	Type of ROC curve.
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".
x1, 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 $b$ considered for the ROC curve estimation: $\hat{a} = \frac{[\bar{\xi}_n - \bar{\chi}_m]}{\hat{s}_\xi}$ and $\hat{b} = \hat{s}_x / \hat{s}_\xi$ .
p0	Estimate of the "central value", $\mu^*$ , about to which the thresholds $x^L$ and $x^U$ are symmetrical.

**References**

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

**Examples**

```
data(HCC)

# ROC curve estimates for gene 03515901 and response tumor assuming the binormal scenario
gROC_param(X = HCC[, "cg03515901"], D = HCC$tumor) # Standard right-sided ROC curve
gROC_param(X = HCC[, "cg03515901"], D = HCC$tumor, side = "left") # Left-sided ROC curve
gROC_param(X = HCC[, "cg03515901"], D = HCC$tumor, side = "both") # gROC curve
```

---

HCC

*Hepatocellular carcinoma data*


---

**Description**

This dataset is derived from gene expression arrays of tumor and adjacent non-tumor tissues of 62 Taiwanese cases of hepatocellular carcinoma. The complete dataset was deposited in [NCBI's Gene Expression Omnibus \(GEO\)](#) and it is available through series accession number GSE37988. This dataset contains 948 from the 27,578 autosomal CpG sites screened.

**Usage**

```
data("HCC")
```

## Format

A data frame with 124 observations on 952 variables. First 4 variables are tissue (identification number for the tissue; from 1 to 62), sex (female or male), age (age in years of the patient), and tumor (status of the tissue; nontumor or tumor). The following 948 from cg03409548 to cg20240860 are numeric variables containing the relative gene expression intensities of the corresponding gene.

## Source

NCBI's Gene Expression Omnibus (GEO) - Series accession number GSE37988 (<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37988>)

## Examples

```
data(HCC)
str(HCC) # Data structure
table(HCC$tumor) # Number of non-tumor and tumor tissues

# Histograms of gene 18384097 expression intensities for non-tumor and tumor tissues
oldpar <- par(mfrow = c(2,1))
x <- subset(HCC, tumor == "nontumor")$cg18384097
y <- subset(HCC, tumor == "tumor")$cg18384097
hist(x, xlim = c(0,1), main = "Gene 18384097 in non-tumor tissues")
hist(y, xlim = c(0,1), main = "Gene 18384097 in tumor tissues")
par(oldpar)
```

---

hROC

*Build a ROC curve for a transformation of a univariate marker*

---

## Description

This is one of the main functions of the movieROC package. It builds a univariate ROC curve for a transformed marker  $h(X)$  and returns a 'hroc' object, a list of class 'hroc'. This object can be [printed](#), [plotted](#), or [predicted](#) for a particular point. It may be also passed to [plot\\_funregions\(\)](#) and [plot\\_regions\(\)](#) functions.

## Usage

```
hROC(X, D, ...)
## Default S3 method:
hROC(X, D, type = c("lrm", "overfitting", "kernel", "h.fun"),
      formula.lrm = "D ~ pol(X,3)", h.fun = function(x) {x}, kernel.h = 1,
      plot.h = FALSE, plot.roc = FALSE, new.window = FALSE,
      main = NULL, xlab = "x", ylab = "h(x)", xaxis = TRUE, ...)
```

**Arguments**

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 <code>lrm()</code> function in <b>rms</b> package), "kernel" (the transformation included in Martínez-Cambor et al. (2021) estimated by the kernel density approach), "overfitting" (the overfitting transformation, $\hat{h}_{of}(\cdot)$ is taken), or "h.fun" (the transformation indicated in the input parameter <code>h.fun</code> 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 <code>density()</code> function in <b>stats</b> package. Default: 1.
h.fun	If type = "h.fun", the transformation employed (as a function in R). Default: <code>function(x){x}</code> .
plot.h	If TRUE, the transformation employed is illustrated.
plot.roc	If TRUE, the resulting ROC curve is illustrated.
new.window	If TRUE, two previous graphics are plotted separately in different windows.
main	A main title for the plot used if <code>plot.h = TRUE</code> .
xlab, ylab	A label for the x and y axis of the plot used if <code>plot.h = TRUE</code> .
xaxis	Graphical parameter used if <code>plot.h = TRUE</code> . If FALSE, plotting of the axis is suppressed.
...	Other parameters to be passed. Not used.

**Details**

A theoretical and practical discussion about the type of transformation considered and its basis may be found in Martínez-Cambor et al. (2019) and Martínez-Cambor et al. (2021).

The overfitting function estimate is defined as follows:

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

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  $B$ ,  $\{y_1, \dots, y_{n_1}\} \subseteq \{\xi_1, \dots, \xi_n\}$  are the positive sample values without ties and  $\{z_1, \dots, z_{n_2}\} \subseteq \{\xi_1, \dots, \xi_n\}$  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.

**Value**

A list of class 'hroc' with the following fields:

levels                      Levels of response values.

X, Y	Original and transformed marker values, respectively.
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.

### Dependencies

If type = "lrm", the lrm() function in the **rms** package is used. This library is also loaded to consider special transformation functions such as pol() and rcs().

### References

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

P. Martínez-Camblor, S. Pérez-Fernández, and S. Díaz-Coto (2021) "Optimal classification scores based on multivariate marker transformations". *ASTA Advances in Statistical Analysis*, **105**(4): 581–599. DOI: [doi:10.1007/s1018202000388z](https://doi.org/10.1007/s1018202000388z).

### Examples

```
data(HCC)

# ROC curve for gene 18384097 to identify tumor by considering 4 different transformations:
X <- HCC$cg18384097; D <- HCC$tumor
## 1. Ordinary cubic polynomial formula for binary logistic regression
hROC(X, D)
## 2. Linear tail-restricted cubic splines for binary logistic regression
hROC(X, D, formula.lrm = "D ~ rcs(X,8)")
## 3. Overfitting transformation for this particular sample
hROC(X, D, type = "overfitting")
## 4. Optimal transformation in terms of likelihood ratio
## by kernel density estimation with bandwidth 3
hROC(X, D, type = "kernel", kernel.h = 3)
```

---

movieROC

*Create a video with the building procedure of the ROC curve*

---

### Description

Save a video as a GIF illustrating the construction of the ROC curve. Particularly, for each element in vector fpr (optional input parameter), the function executed is

- for 'groc' objects:

```
plot_buildROC(obj, FPR = fpr[i], build.process = TRUE, ...);
```

- for 'multiroc' objects:

```
plot_buildROC(obj, FPR = fpr[i], build.process = TRUE, display.method, displayOV, ...)
```



**Usage**

```

movieROC(x, ...)
## S3 method for class 'groc'
movieROC(x, fpr = NULL, h = c(1,1), histogram = FALSE, breaks = 15,
  reduce = TRUE, completeROC = FALSE, videobar = TRUE, file = "animation1.gif",
  save = TRUE, legends = FALSE, speedcorrection = FALSE, tpause = 1, interval = 0.2,
  ani.width, ani.height, xlab = "Marker", main.density = "Density functions",
  cex.lab = 2.5, cex.axis = 1.75, cex.main = 2.25 + as.numeric(reduce),
  xlim = NULL, ylim = NULL, cex.point = 1.5, lwd.curve = 2, mar = NULL,
  lim.density = 0.01, col.controlscases = c("#485C99", "#8F3D52"),
  col.curve = "black", col.threshold = "#FCBA04", verbose = FALSE, ...)
## S3 method for class 'multiroc'
movieROC(x, fpr = NULL, display.method = c("PCA", "OV"),
  displayOV = c(1,2), border = TRUE, completeROC = FALSE, videobar = TRUE,
  file = "animation1.gif", save = TRUE, legends = FALSE, tpause = 1,
  interval = 0.2, ani.width, ani.height, xlab = NULL, ylab = NULL,
  cex = 0.8, cex.lab = 1.5, cex.axis = 1.5, cex.main = 2, alpha.points = 1,
  alpha.contour = 0.25, lwd.curve = 2, lty.curve = 1, lf = NULL,
  col.controlscases = c('#485C99', '#8F3D52'), col.curve = 'black',
  col.threshold = '#FCBA04', verbose = FALSE, ...)

```

**Arguments**

x	An ROC curve object from the <b>movieROC</b> package. Possible classes are: 'groc' (output of <code>gROC()</code> function) and 'multiroc' (output of <code>multiROC()</code> function).
fpr	A vector of false-positive rates for which the building process of the ROC curve is displayed. Default: if length of <code>x\$t</code> is lower than 150, such vector is taken as <code>fpr</code> ; otherwise, an equally-space vector of length 100 covering the range of the marker is considered.
h	A vector of length 2 with the bandwidth used to compute kernel density estimation for controls and cases, respectively. See <code>adjust</code> parameter in <code>density()</code> function from <b>stats</b> package. Default: <code>h = c(1,1)</code> .
histogram	If TRUE, histograms are displayed on the left instead of kernel density estimates. Default: FALSE. Only available for a 'groc' object.
breaks	If <code>histogram = TRUE</code> , number of breaks used for the histograms. Default: 15.
reduce	If FALSE, two extra graphics are displayed at the bottom (see Details for more information). Default: TRUE. Only available for a 'groc' object.
completeROC	A logical value indicating if the whole ROC curve should be displayed in light gray or not. Default: FALSE.
videobar	If TRUE, a text progress bar is shown in the R console. Default: TRUE.
verbose	If TRUE, a progress bar is displayed for computationally intensive methods. Default: FALSE.
file	File name of the movie (with the extension). Default: "animation1.gif".
save	If TRUE, video is saved as a GIF by using the <code>saveGIF</code> function in <b>animation</b> package. Default: TRUE.

legends	If TRUE, a legend with colors meaning is displayed on the left graphic and the AUC is shown on the right graphic. Default: FALSE.
speedcorrection	If TRUE, only some FPR in fpr vector are considered to make the video faster. Default: FALSE. Only available for a 'groc' object.
tpause	If save = FALSE, time interval to suspend execution for, in seconds. Default: 1.
interval	If save = TRUE, a positive number to set the time interval of the animation (unit in seconds) in <b>animation</b> package. Default: 0.2.
ani.width, ani.height	If save = TRUE, width and height of image frames (unit in px) in <b>animation</b> package.
xlab, ylab	Label for x- and y-axis on the left plot. ylab not available for object of class 'groc'.
main.density	Title for the left plot. Only available for a 'groc' object.
cex.lab, cex.axis, cex.main	The magnification to be used for labels, axis annotation and main titles, respectively, relative to the current setting of cex.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default. Default: 0.8. Not available for object of class 'groc' (cex=1 in this case).
cex.point	The magnification to be used for the particular point on the ROC curve and its text, relative to the current setting of cex. Default: cex.point = 1.5. Only available for a 'groc' object.
xlim, ylim	Range for x- and y-axis on the left plot. Only available for a 'groc' object.
lim.density	If xlim is NULL, lower limit for the density estimate to compute the range for the x-axis on the left plot. Default: 0.01. Only available for a 'groc' object.
lty.curve, lwd.curve, col.curve	The line type, width and color for ROC curve. Default: lty.curve = 1 (solid), lwd.curve = 2 and col.curve = "black".
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. Only available for a 'groc' object.
col.controlscales	Vector of length 2 with the color used to control and case group, respectively. Default: c('#485C99', '#8F3D52').
col.threshold	Color for the chosen point. Default: "FCBA04".
border	If TRUE, a border for the classification subsets is drawn. Default: TRUE. Not available for object of class 'groc'.
alpha.points, alpha.contour	Number in [0,1] modifying the opacity alpha of the color for the points and classification region. The adjustcolor function in <b>grDevices</b> package is used. Default: alpha.points = 1, alpha.contour = 0.25. Not available for object of class 'groc'.
lf	Epsilon value for steps. Not used.

`display.method` Method to display the marker values from a 'multiroc' object on the left plot. Methods available: "OV" (projected over two selected components of the marker indicated in `displayOV`), or "PCA" (projected over the two principal components from a Principal Component Analysis). Default: "PCA". Only available for object of class 'multiroc'.

`displayOV` If `display.method = "OV"`, the two components of the marker used to project the marker values on the left plot. Default: `c(1,2)` (two first components). Only available for object of class 'multiroc'.

... Other parameters to be passed to the `saveGIF()` function in **animation** package.

### Value

A video with the building procedure of the ROC curve estimate with the selected graphical parameters

### Dependencies

If `save = TRUE` (by default), the `saveGIF()` function in the **animation** package is used.

### Examples

```
data(HCC)

# Standard ROC curve for gene 20202438
roc_cg20202438 <- gROC(X = HCC$cg20202438, D = HCC$tumor, side = "right")

### The video will be saved as a GIF with the name "video_cg20202438"
movieROC(roc_cg20202438, file = "video_cg20202438.gif", save = FALSE)

# Multivariate ROC curve for genes 0202438, 18384097, and 03515901
multiroc_PT <- multiROC(X = cbind(HCC$cg20202438, HCC$cg18384097, HCC$cg03515901),
  D = HCC$tumor, method = "fixedLinear", methodLinear = "PepeThompson")

# Two first components from PCA:
### The video will be saved as a GIF with the name "video_multiroc_pca"
movieROC(multiroc_PT, file = "video_multiroc_pca.gif", save = FALSE)

### The video will be saved as a GIF with the name "video_multiroc_orig"
movieROC(multiroc_PT, display.method = "OV", displayOV = c(1,3),
  file = "video_multiroc_orig.gif", cex = 1.2, xlab = "Gene 20202438",
  ylab = "Gene 03515901", lwd.curve = 4, save = FALSE)
```

---

`movieROC2_densities` *Create a video with the building procedure of the smooth ROC curve estimate*

---

## Description

This function tracks the construction of the standard ROC curve (right- or left-sided, depending on the side of the object of class 'groc' included) resulting from the kernel density function estimation for controls and cases. Four graphics are displayed: top-left, the kernel density estimates; top-right, the resulting ROC curve; bottom-left, boxplots and points for controls and cases and classification subset in gray color; bottom-right, classification subsets for every FPR until the current one. It makes use of the `plot_densityROC()` function for each screenshot.

## Usage

```
movieROC2_densities(obj, h = c(1, 1), cut.off = NULL, completeROC = FALSE,
  legends = FALSE, videobar = TRUE, file = "animation1.gif", clean = FALSE,
  interval = 0.2, ani.width = 500, ani.height = 750, save = TRUE,
  tpause = 1, verbose = FALSE, ...)
```

## Arguments

<code>obj</code>	An object of class 'groc' with <code>side = "right" or "left"</code> .
<code>h</code>	A vector of length 2 with the bandwidth used to compute kernel density estimation for controls and cases, respectively. See <code>adjust</code> parameter in <code>density()</code> function from <b>stats</b> package. Default: <code>h = c(1, 1)</code> .
<code>cut.off</code>	Vector with marker cutoffs for which the graphics are displayed. Default: number of unique marker values is lower than 150, these are considered; otherwise, a equally-spaced grid of length 102 in the range of the marker is used.
<code>save</code>	If TRUE, video is saved as a GIF by using the <code>saveGIF()</code> function in <b>animation</b> package. Default: TRUE.
<code>completeROC</code>	A logical value indicating if the whole ROC curve should be displayed in light gray or not. Default: FALSE.
<code>legends</code>	If TRUE, legends with the meaning of colors are displayed. Default: FALSE.
<code>videobar</code>	If TRUE, a text progress bar is shown in the R console. Default: TRUE.
<code>file</code>	File name of the movie (with the extension). Default: <code>"animation1.gif"</code> .
<code>clean</code>	Whether to delete the individual image frames in <b>animation</b> package. Default: FALSE.
<code>interval</code>	A positive number to set the time interval of the animation (unit in seconds) in <b>animation</b> package. Default: 0.2.
<code>ani.width, ani.height</code>	Width and height of image frames (unit in px) in <b>animation</b> package.
<code>tpause</code>	If <code>save = FALSE</code> , time interval to suspend execution for, in seconds. Default: 1.
<code>verbose</code>	If TRUE, a progress bar is displayed for computationally intensive methods. Default: FALSE.
<code>...</code>	Other parameters to be passed to the <code>saveGIF</code> function in <b>animation</b> package.

## Value

A video with the building procedure of the smooth ROC curve estimate with the selected graphical parameters

## Dependencies

If `save = TRUE` (by default), the `saveGIF()` function in the **animation** package is used.

## Examples

```
data(HCC)

# Standard ROC curve for gene 20202438
roc_cg20202438 <- gROC(X = HCC$cg20202438, D = HCC$tumor)

### The video will be saved as a GIF with the name "video_cg20202438_smooth"
movieROC2_densities(roc_cg20202438, file = "video_cg20202438_smooth.gif", save = FALSE)
```

---

multiROC

*Build a ROC curve for a multivariate marker with dimension  $p$*

---

## Description

This is one of the main functions of the `movieROC` package. It builds a multivariate ROC curve by considering one of these methods: i) fitting a binary logistic regression model with a particular combination (fixed by the user) of the two components on the right-hand side, ii) linear combinations with fixed parameters, or iii) linear combinations with dynamic parameters, or iv) estimating optimal transformation based on kernel density estimation, or v) quadratic combinations with fixed parameters (if  $p = 2$ ). It returns a ‘multiroc’ object, a list of class ‘multiroc’. This object can be [printed](#) or [plotted](#). It may be also passed to [plot\\_buildROC\(\)](#) and [movieROC\(\)](#) function.

## Usage

```
multiROC(X, D, ...)
## Default S3 method:
multiROC(X, D,
  method = c("lrm", "fixedLinear", "fixedQuadratic", "dynamicEmpirical",
             "dynamicMeisner", "kernelOptimal"),
  formula.lrm = "D ~ X.1 + I(X.1^2) + X.2 + I(X.2^2) + I(X.1*X.2)",
  stepModel = TRUE,
  methodLinear = c("coefLinear", "SuLiu", "PepeThompson", "logistic", "minmax"),
  coefLinear = rep(1, ncol(X)), coefQuadratic = c(1, 1, 0, 1, 1),
  K = 201, alpha = 0.5, approxh = 0.5, multiplier = 2,
  kernelOptimal.H = c("Hbcv", "Hscv", "Hpi", "Hns", "Hlscv", "Hbcv.diag",
                     "Hscv.diag", "Hpi.diag", "Hlscv.diag"),
  eps = sqrt(.Machine$double.eps), verbose = FALSE, ...)
```

## Arguments

**X** Matrix (dimension  $n \times p$ ) of marker values where  $n$  is the sample size and  $p$  is the dimension of the multivariate marker.

**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), "fixedLinear" (linear frontiers with fixed parameters given in coefLinear or estimated by the 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), "dynamicEmpirical" (linear frontiers with dynamic parameters reported by the empirical method, only available for $p = 2$ ), or "kernelOptimal" (estimating optimal transformation based on bivariate kernel density estimation by Martínez-Camblor et al. (2021) using the kde() function in the <b>ks</b> package). Default: "lrm".
formula.lrm	If method = "lrm", the transformation employed in the right-hand side of the logistic regression model (in terms of $X.1$ , $X.2$ dots, $X.p$ , and $D$ ). Default: quadratic formula for the two first components $X.1$ and $X.2$ .
stepModel	If TRUE and method = "lrm", a model selection is performed based on the AIC (Akaike information criterion) in a stepwise algorithm (see step() function in <b>stats</b> package for more information). Default: TRUE.
methodLinear	If method = "fixedLinear", method used to build the classification regions. One of "coefLinear" (particular fixed coefficients in coefLinear), "SuLiu" (Su and Liu, 1993), "PepeThompson" (Pepe and Thompson, 2000), "logistic" (logistic regression model), "minmax" (Liu et al., 2011). Default: "coefLinear".
coefLinear	If method = "fixedLinear" and methodLinear = "coefLinear", a vector of length $p$ with the coefficients $\beta_i$ ( $i \in \{1, \dots, p\}$ ) used to $\mathcal{L}_\beta(\mathbf{X}) = \beta_1 X_1 + \dots + \beta_p X_p$ . Default: (1, ..., 1).
coefQuadratic	If method = "fixedQuadratic", a vector of length 5 with coefficients $\beta_1, \dots, \beta_5$ used to $\mathcal{Q}_\beta(\mathbf{X}) = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2$ . Default: (1, 1, 0, 1, 1).
alpha, approxh, multiplier	If method = "dynamicMeisner", input parameters used in the maxTPR() function of the <b>maxTPR</b> package (internally integrated because the library is no longer available in CRAN). 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.
kernelOptimal.H	If method = "kernelOptimal", the bandwidth matrix H used in the kde() function of the <b>ks</b> package. Default: "Hbcv" (biased cross-validation (BCV) bandwidth matrix selector for bivariate data) if $p = 2$ , "Hpi" (plug-in bandwidth selector) if $p > 2$ .
eps	Epsilon value to consider. Default: sqrt(.Machine\$double.eps).
verbose	If TRUE, a progress bar is displayed for computationally intensive methods. Default: FALSE.
...	Other parameters to be passed. Not used.

**Value**

A list of class ‘multiroc’ with the following fields:

controls, cases	Marker values of negative and positive subjects, respectively.
levels	Levels of response values.
t	Vector of false-positive rates.
roc	Vector of values of the ROC curve for t.
auc	Area under the curve estimate.
Z	If method $\neq$ "dynamicMeisner" and method $\neq$ "dynamicEmpirical", resulting univariate marker values.
c	If method $\neq$ "dynamicMeisner" and method $\neq$ "dynamicEmpirical", vector of final marker thresholds resulting in (t, roc).
CoefTable	If method = "dynamicMeisner" or "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 $100 \times 100$ with the marker values over a grid of $(X_1, X_2)$ bivariate values (f).

**Dependencies**

If method = "lrm", the glm() function in the **stats** package is used.

If method = "kernelOptimal", the kde() function in the **ks** package is used.

**References**

- J. Q. Su and J. S. Liu. (1993) "Linear combinations of multiple diagnostic markers". *Journal of the American Statistical Association*, **88**(424): 1350–1355. DOI: [doi:10.1080/01621459.1993.10476417](https://doi.org/10.1080/01621459.1993.10476417).
- M. S. Pepe and M. L. Thompson (2000) "Combining diagnostic test results to increase accuracy". *Biostatistics*, **1** (2):123–140. DOI: [doi:10.1093/biostatistics/1.2.123](https://doi.org/10.1093/biostatistics/1.2.123).
- C. Liu, A. Liu, and S. Halabi (2011) "A min–max combination of biomarkers to improve diagnostic accuracy". *Statistics in Medicine*, **30**(16): 2005–2014. DOI: [doi:10.1002/sim.4238](https://doi.org/10.1002/sim.4238).
- P. Martínez-Camblor, S. Pérez-Fernández, and S. Díaz-Coto (2021) "Optimal classification scores based on multivariate marker transformations". *AStA Advances in Statistical Analysis*, **105**(4): 581–599. DOI: [doi:10.1007/s1018202000388z](https://doi.org/10.1007/s1018202000388z).
- A. Meisner, M. Carone, M. S. Pepe, and K. F. Kerr (2021) "Combining biomarkers by maximizing the true positive rate for a fixed false positive rate". *Biometrical Journal*, **63**(6): 1223–1240. DOI: [doi:10.1002/bimj.202000210](https://doi.org/10.1002/bimj.202000210).

**Examples**

```
data(HCC)

# ROC curve for genes 20202438 and 18384097 (p=2) to identify tumor by 4 different methods:
X <- cbind(HCC$cg20202438, HCC$cg18384097); D <- HCC$tumor
## 1. Linear combinations with fixed parameters by Pepe and Thompson (2000)
```

```

multiROC(X, D, method = "fixedLinear", methodLinear = "PepeThompson")
## 2.Linear combinations with dynamic parameters by Meisner et al. (2021)
### Time consuming
multiROC(X, D, method = "dynamicMeisner")
## 3. Logistic regression model with quadratic formula by default
multiROC(X, D)
## 4. Optimal transformation with multivariate KDE by Martínez-Camblor et al. (2021)
multiROC(X, D, method = "kernelOptimal")

# ROC curve for genes 20202438, 18384097, and 03515901 (p=3) to identify tumor
# by 4 different methods:
X <- cbind(HCC$cg20202438, HCC$cg18384097, HCC$cg03515901); D <- HCC$tumor
## 1. Linear combinations with fixed parameters by Pepe and Thompson (2000)
multiROC(X, D, method = "fixedLinear", methodLinear = "PepeThompson")
## 2.Linear combinations with dynamic parameters by Meisner et al. (2021)
### Time consuming
multiROC(X, D, method = "dynamicMeisner")
## 3. Logistic regression model with quadratic formula by default
multiROC(X, D)
## 4. Optimal transformation with multivariate KDE by Martínez-Camblor et al. (2021)
multiROC(X, D, method = "kernelOptimal")

```

---

plot

*Plot an ROC curve*


---

## Description

This is one of the core functions of the movieROC package. It displays the empirical ROC curve estimate from an object of class ‘groc’, ‘hroc’, or ‘multiroc’.

## Usage

```

## S3 method for class 'groc'
plot(x, xlim = c(0, 1), ylim = c(0, 1), lwd = 3,
     xlab = "False-Positive Rate", ylab = "True-Positive Rate", main = "ROC curve",
     cex.lab = 1.25, cex.main = 1.5, type = NULL, new = TRUE, ...)
## S3 method for class 'hroc'
plot(x, type = 'S', xlim = c(0,1), ylim = c(0,1),
     lwd = 3, xlab = "False-Positive Rate", ylab = "True-Positive Rate",
     main = "ROC Curve", cex.lab = 1.25, cex.main = 1.5, new = TRUE, ...)
## S3 method for class 'multiroc'
plot(x, ...)

```

## Arguments

x	An ROC curve object from movieROC package. Possible classes are: ‘groc’ (output of <code>gROC()</code> function), ‘hroc’ (output of <code>hROC()</code> function), and ‘multiroc’ (output of <code>multiROC()</code> function).
xlim, ylim	Range for x- and y-axis. Default: unit interval.



lwd	Line width of the ROC curve. Default: 3.
xlab, ylab	Label for x- and y-axis.
main	Title for the plot.
cex.lab, cex.main	The magnification to be used for labels and main title, respectively, relative to the current setting of cex. Default: cex.lab = 1.25, cex.main = 1.5.
type	What type of plot should be drawn (see help from plot function in <b>base</b> package). Default: type = "s" (stair steps), except for x object from gROC_param function, in which case type = "l" (lines).
new	If TRUE, a new plot is displayed; otherwise, the ROC curve is plotted over the existing graphic. Default: TRUE.
...	Other graphical parameters to be passed.

### Value

A plot of the ROC curve with the selected graphical parameters

### Examples

```

data(HCC)
# ROC curve estimates for gene 03515901 and response tumor
rroc <- gROC(X = HCC[, "cg03515901"], D = HCC$tumor) # Right-sided
lroc <- gROC(X = HCC[, "cg03515901"], D = HCC$tumor, side = "left") # Left-sided
hroc <- hROC(X = HCC[, "cg03515901"], D = HCC$tumor) # Transformed by a cubic polinomial

plot(rroc, lty = 2, frame = FALSE)
plot(lroc, new = FALSE)
plot(hroc, new = FALSE, col = "blue")
legend("topleft", legend = c("Right-sided", "Left-sided", "Transformed marker"),
      col = c("black", "black", "blue"), lty = c(1,2,1), lwd = 2, bty = "n")

# ROC curve estimate for genes 20202438 and 18384097 to simultaneously identify tumor
# by a logistic regression model with quadratic formula
biroc <- multiROC(X = cbind(HCC$cg20202438, HCC$cg18384097), D = HCC$tumor)
plot(biroc)
legend("bottomright", paste("AUC = ", format(biroc$auc, digits = 3)))

# ROC curve estimate for genes 20202438, 18384097 and 03515901 to simultaneously
# identify tumor by a linear combinations with fixed parameters by Pepe and Thompson (2000)
multiroc <- multiROC(X = cbind(HCC$cg20202438, HCC$cg18384097, HCC$cg03515901),
  D = HCC$tumor, method = "fixedLinear", methodLinear = "PepeThompson")
plot(multiroc)
legend("bottomright", paste("AUC = ", format(multiroc$auc, digits = 3)))

```

plot\_buildROC

*Plot the building procedure of the ROC curve***Description**

This function tracks the ROC curve. It plots two graphics in the same figure: right, the empirical ROC curve estimate; left, classification subset for a particular FPR or threshold(s) for the marker (threshold only for object of class 'groc').

**Usage**

```
## S3 method for class 'groc'
plot_buildROC(x, FPR = NULL, C = NULL, XL = NULL, XU = NULL, h = c(1,1),
  histogram = FALSE, breaks = 15, reduce = TRUE, build.process = FALSE,
  completeROC = FALSE, new.window = FALSE, legends = FALSE, type = 's',
  cex.point = 1.5, lwd.curve = 2, mar = NULL, lim.density = 0.01, xlim = NULL,
  ylim = NULL, cex.lab = 1.5, cex.axis = 1.5, cex.main = 2, xlab = "Marker",
  main.density = "Density functions", col.controlscases = c('#485C99', '#8F3D52'),
  col.threshold = '#FCBA04', col.curve = 'black', eps = sqrt(.Machine$double.eps),
  ...)
## S3 method for class 'multiroc'
plot_buildROC(x, FPR = 0.15, display.method = c("PCA", "OV"),
  displayOV = c(1,2), build.process = FALSE, completeROC = TRUE,
  new = FALSE, new.window = FALSE, border = FALSE, cutoff = TRUE, legends = FALSE,
  type = 's', col.controlscases = c('#485C99', '#8F3D52'),
  col.threshold = '#FCBA04', col.curve = 'black', cex.point = 1.5,
  alpha.points = .75, alpha.contour = 0.25, lwd.curve = 2, lty.curve = 1,
  cex = 0.8, cex.lab = 1.5, cex.axis = 1.5, cex.main = 2, xlab = NULL, ylab = NULL,
  lf = NULL, eps = sqrt(.Machine$double.eps), ...)
```

**Arguments**

x	An ROC curve object from movieROC package. Possible classes are: 'groc' (output of <code>gROC()</code> function) and 'multiroc' (output of <code>multiROC()</code> function).
FPR	False-positive rate for which the left plot is computed. Default: 0.15.
C	Marker cutoff for which the left plot is computed. Only available for a 'groc' object.
XL, XU	Marker cutoffs for which the left plot is computed. Only available for object of class 'groc' with <code>side = "both"</code> or <code>"both2"</code> .
h	A vector of length 2 with the bandwidth used to compute kernel density estimation for controls and cases, respectively. See <code>adjust</code> parameter in <code>density()</code> function from <code>stats</code> package. Default: <code>h = c(1,1)</code> .
histogram	If TRUE, histograms are displayed on the left instead of kernel density estimates. Default: FALSE. Only available for a 'groc' object.
breaks	If <code>histogram = TRUE</code> , number of breaks used for the histograms. Default: 15.

reduce	If FALSE, two extra graphics are displayed at the bottom (see Details for more information). Default: TRUE. Only available for a 'groc' object.
build.process	If FALSE, the whole ROC curve is displayed; otherwise, if completeROC = TRUE, the portion of the ROC curve until the fixed FPR is highlighted in black and the rest is shown in gray. Default: FALSE.
completeROC	If build.process = TRUE, a logical value indicating if the whole ROC curve should be displayed in light gray or not. Default: FALSE for 'groc' object; TRUE for 'multiroc' object.
new.window, new	If TRUE, graphics are displayed in a new window. Default: FALSE. new not available for object of class 'groc'.
legends	If TRUE, a legend with colors meaning is displayed on the left graphic and the AUC is shown on the right graphic. Default: FALSE.
type, lty.curve, lwd.curve, col.curve	The line type, width and color for ROC curve. Default: type = "s" (stair steps), lty.curve = 1 (solid), lwd.curve = 2 and col.curve = "black".
cex.lab, cex.axis, cex.main	The magnification to be used for labels, axis annotation and main titles, respectively, relative to the current setting of cex. Default: cex.lab = 1.5, cex.axis = 1.5, cex.main = 2.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default. Default: 0.8. Not available for object of class 'groc' (cex=1 in this case).
cex.point	The magnification to be used for the particular point on the ROC curve and its text, relative to the current setting of cex. Default: cex.point = 1.5.
xlim, ylim	Range for x- and y-axis on the left plot. Only available for a 'groc' object.
lim.density	If xlim is NULL, lower limit for the density estimate to compute the range for the x-axis on the left plot. Default: 0.01. Only available for a 'groc' object.
xlab, ylab	Label for x- and y-axis on the left plot. ylab not available for object of class 'groc'.
main.density	Title for the left plot. Only available for a 'groc' object.
col.controlscases	Vector of length 2 with the color used to control and case group, respectively. Default: c('#485C99', '#8F3D52').
col.threshold	Color for the chosen point. Default: "FCBA04".
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. Only available for a 'groc' object.
eps	Epsilon value to consider. Default: sqrt(.Machine\$double.eps).
border	If TRUE, a border for the classification subsets is drawn. Default: FALSE for 'multiroc' object (TRUE if $p = 2$ ). Not available for object of class 'groc'.
cutoff	If TRUE, the cutoff value for the resulting univariate marker is displayed on the right plot (ROC curve). Default: TRUE. Not available for object of class 'groc'.

alpha.points, alpha.contour	Number in [0,1] modifying the opacity alpha of the color for the points and classification region. The adjustcolor() function in <b>grDevices</b> package is used. Default: alpha.points = 0.75, alpha.contour = 0.25. Not available for object of class 'groc'.
lf	Epsilon value for steps. Not used. Not available for object of class 'groc'.
display.method	Method to display the marker values from a 'multiroc' object on the left plot. Methods available: "OV" (projected over two selected components of the marker indicated in displayOV), or "PCA" (projected over the two principal components from a Principal Component Analysis). Default: "PCA". Only available for object of class 'multiroc'.
displayOV	If display.method = "OV", the two components of the marker used to project the marker values on the left plot. Default: c(1, 2) (two first components). Only available for object of class 'multiroc'.
...	Other parameters to be passed. Not used.

## Details

- For univariate ROC curves:

It plots two graphics in the same figure: left, density function estimates for the marker in both populations with the areas corresponding to FPR and TPR colored (blue and red, respectively) for a particular FPR or threshold(s) for the marker; right, the empirical ROC curve estimate.

For an object of class 'groc', if the input parameter reduce = FALSE, two more graphics are displayed at the bottom: left, boxplots for the marker distribution in both populations; right, classification subsets for every false-positive rate (grey color by default).

- For multivariate ROC curves:

Plot two graphics in the same figure: right, the ROC curve highlighting the point and the threshold for the resulting univariate marker; left, scatterplot with the marker values for controls (blue color) and cases (red color), and the classification subset (in gold color) reporting the false-positive rate selected by the user (if FPR is not NULL).

If build.process = FALSE, the whole ROC curve is displayed; otherwise, if completeROC = TRUE, the portion of the ROC curve until the fixed FPR is highlighted in black and the rest is shown in gray, while if completeROC = FALSE, only the first portion of the curve is illustrated.

## Value

A plot of the building procedure of the ROC curve with the selected graphical parameters

## Dependencies

If method = "kernelOptimal" in a 'multiroc' object, the na.locf() function in the **zoo** package is used.

**Examples**

```

data(HCC)

# Standard ROC curve for gene 20202438
roc_cg20202438 <- gROC(X = HCC$cg20202438, D = HCC$tumor, side = "right")
plot_buildROC(roc_cg20202438)
plot_buildROC(roc_cg20202438, C = .77, build.process = TRUE, reduce = FALSE)

# Multivariate ROC curve for genes 0202438, 18384097, and 03515901
multiroc_PT <- multiROC(X = cbind(HCC$cg20202438, HCC$cg18384097, HCC$cg03515901),
  D = HCC$tumor, method = "fixedLinear", methodLinear = "PepeThompson")
plot_buildROC(multiroc_PT, cex = 1.2, lwd.curve = 4) # Two first components from PCA
plot_buildROC(multiroc_PT, display.method = "OV", displayOV = c(1,3),
  cex = 1.2, xlab = "Gene 20202438", ylab = "Gene 03515901", lwd.curve = 4)

```

plot\_densities

*Plot density function estimates for controls and cases***Description**

This function plots the kernel density function estimates for controls and cases from an object of class 'groc'. It offers the possibility of plotting the histograms instead of the smooth estimates.

**Usage**

```

plot_densities(obj, h = c(1, 1), histogram = FALSE, breaks = 15,
  col = c("#485C99", "#8F3D52"), lwd = 2, xlim = NULL, ylim = NULL,
  xaxs = "i", yaxs = "i", xlab = "Marker", ylab = "f(x)",
  main = "Density functions", legend = FALSE, pos.legend = "topright",
  cex.lab = 1.5, cex.axis = 1.5, cex.main = 2, cex.legend = 1,
  eps = sqrt(.Machine$double.eps), new = TRUE, ...)

```

**Arguments**

obj	An object of class 'groc'.
h	A vector of length 2 with the bandwidth used to compute kernel density estimation for controls and cases, respectively. See adjust parameter in density() function from <b>stats</b> package. Default: h = c(1,1).
histogram	If TRUE, histograms are displayed instead of kernel density estimates. Default: FALSE.
breaks	If histogram = TRUE, number of breaks used for the histograms. Default: 15.
col	A vector of length 2 with color used for controls and cases, respectively. Default: c("#485C99", "#8F3D52").
lwd	Line width for the density function or histogram. Default: 2.
xlim, ylim	Range for x- and y-axis.

<code>xaxs, yaxs</code>	The style of axis interval calculation to be used for the x- and the y-axis, respectively. Default: "i" (internal; just finds an axis with pretty labels that fits within the original data range). For more information, see help for <code>par</code> .
<code>xlab, ylab</code>	Label for x- and y-axis.
<code>main</code>	Title for the plot.
<code>cex.lab, cex.axis, cex.main</code>	The magnification to be used for labels, axis annotation and main titles, respectively, relative to the current setting of <code>cex</code> . Default: <code>cex.lab = 1.5</code> , <code>cex.axis = 1.5</code> , <code>cex.main = 2</code> .
<code>legend</code>	If TRUE, a legend with the meaning of colors is displayed on the left plot. Default: FALSE.
<code>pos.legend, cex.legend</code>	The position and magnification to be used for legend, relative to the current setting of <code>cex</code> . Default: <code>pos.legend = "topright"</code> , <code>cex.legend = 1</code> .
<code>eps</code>	Epsilon value to consider. Default: <code>sqrt(.Machine\$double.eps)</code> .
<code>new</code>	If TRUE, a new plot is displayed; otherwise, density estimates are displayed over the current plot. Default: TRUE.
<code>...</code>	Other parameters to be passed. Not used.

**Value**

A plot of the kernel density function estimates for both populations with the selected graphical parameters

**Examples**

```
data(HCC)
roc_cg20202438 <- gROC(X = HCC$cg20202438, D = HCC$tumor)
plot_densities(roc_cg20202438, main = "Density functions and histogram")
plot_densities(roc_cg20202438, histogram = TRUE, new = FALSE)
```

---

plot\_densityROC

*Plot standard smooth ROC curve estimate*

---

**Description**

This function estimates the standard ROC curve (right- or left-sided, depending on the side of the object of class 'groc' included) resulting from the kernel density function estimation for controls and cases. Two graphics are displayed: left, the kernel density estimates; right, the resulting ROC curve. Two extra graphics may be shown at the bottom if the user introduces a value for the input parameter `C`: left, boxplots and points for controls and cases and corresponding classification subset in gray color; right, classification subsets for every FPR until the one corresponding to the chosen `C`.

**Usage**

```
plot_densityROC(obj, h = c(1, 1), C = NULL, build.process = FALSE,
  completeROC = TRUE, legends = FALSE, rel.tol = 0.001,
  par.specify = FALSE, cex.lab = 1.5, cex.axis = 1.25, cex.main = 1.75,
  lwd = 2, col = c("#485C99", "#8F3D52"), col.roc = "blue", ...)
```

**Arguments**

obj	An object of class 'groc' with side = "right" or "left".
h	A vector of length 2 with the bandwidth used to compute kernel density estimation for controls and cases, respectively. See adjust parameter in density() function from <b>stats</b> package. Default: h = c(1,1).
C	Marker cutoff for which the graphics are displayed. Default: none.
build.process	If FALSE, the whole ROC curve is displayed; otherwise, if completeROC = TRUE, the portion of the ROC curve until the fixed FPR (resulting from the chosen C) is highlighted in black and the rest is shown in gray. Default: FALSE.
completeROC	If build.process = TRUE, a logical value indicating if the whole ROC curve should be displayed in light gray or not. Default: TRUE.
legends	If TRUE, legends with the meaning of colors are displayed. Default: FALSE.
rel.tol	Relative accuracy requested for the integrate() function from <b>stats</b> package. Default: 0.001.
par.specify	If FALSE, graphics are organized in one row and two columns. Default: FALSE.
cex.lab, cex.axis, cex.main	The magnification to be used for labels, axis annotation and main titles, respectively, relative to the current setting of cex. Default: cex.lab = 1.5, cex.axis = 1.25, cex.main = 1.75.
lwd, col.roc	Line width and color for the ROC curve. Default: lwd = 2, col.roc = "blue".
col	A vector of length 2 with color used for controls and cases, respectively. Default: c("#485C99", "#8F3D52").
...	Other parameters to be passed. Not used.

**Value**

A plot of the standard smooth ROC curve estimate with the selected graphical parameters

**Examples**

```
data(HCC)
roc_cg20202438 <- gROC(X = HCC$cg20202438, D = HCC$tumor)
plot_densityROC(roc_cg20202438)
plot_densityROC(roc_cg20202438, h = c(2,2))
```

---

plot\_funregions      *Plot the transformation function used for the marker*

---

### Description

This function plots the transformation function used for the marker for an object of class ‘hroc’ or class ‘groc’ (this one is only allowed for objects with self-contained classification subsets). Over this graphic, the classification region for a particular FPR chosen by the user is displayed in blue color. If the user specifies a second FPR (FPR2), the classification region is displayed on the same graphic in red color.

### Usage

```
## S3 method for class 'hroc'
plot_funregions(x, FPR = 0.15, FPR2 = NULL,
  plot.subsets = TRUE, new.window = FALSE, main = NULL, ylim = NULL, ...)
## S3 method for class 'groc'
plot_funregions(x, FPR = 0.15, FPR2 = NULL,
  plot.subsets = TRUE, new.window = FALSE, main = NULL, ylim = NULL, ...)
```

### Arguments

x	An object of class ‘hroc’ or ‘groc’.
FPR	False-positive rate used to display the classification region in blue. Default: 0.15.
FPR2	Other false-positive rate used to display the classification region in red. Default: none.
plot.subsets	If TRUE, the classification subsets are displayed. Otherwise, only the transformation function. Default: TRUE.
new.window	If TRUE, graphics are displayed in a new window. Default: FALSE.
main	Title for the plot.
ylim	Range for the y-axis.
...	Other parameters to be passed. Not used.

### Value

A plot of the transformation function used for the marker with the selected graphical parameters

### Examples

```
data(HCC)

hroc_cg18384097 <- hROC(X = HCC$cg18384097, D = HCC$tumor,
  formula.lrm = "D ~ rcs(X,8)")
plot_funregions(hroc_cg18384097)
plot_funregions(hroc_cg18384097, FPR = .1, FPR2 = .5)
```



---

plot\_regions *Plot the classification regions underlying a ROC curve*

---

### Description

This function plots the classification regions for univariate markers. It works for objects of class ‘groc’ and ‘hroc’. Two graphics are displayed in the same figure: left, classification subsets for every false-positive rate (grey color by default); right, 90° rotated ROC curve.

### Usage

```
## S3 method for class 'groc'
plot_regions(x, FPR = 0.15, plot.roc = TRUE, plot.auc = FALSE,
  col = c("white", "grey"), col.FPR = "blue", lwd = 2, new.window = TRUE,
  type.plotroc = "s", xlim = NULL, mar = c(5, 6, 4, 0.25),
  cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.75, main = NULL,
  xlab = "", ylab = "False-Positive Rate", main.plotroc = "ROC curve",
  legend = TRUE, cex.legend = 1, ...)
## S3 method for class 'hroc'
plot_regions(x, FPR = 0.15, plot.roc = TRUE, plot.auc = FALSE,
  col = c('white','grey'), col.FPR = 'blue', lwd = 2, new.window = TRUE,
  type.plotroc = 's', xlim = NULL, mar = c(5,6,4,0.25),
  cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.75, main = NULL,
  xlab = "", ylab = "False-Positive Rate", main.plotroc = "ROC curve",
  legend = TRUE, cex.legend = 1, verbose = FALSE, ...)
```

### Arguments

x	An ROC curve object from a univariate marker. Possible classes are: ‘groc’ (output of gROC function) and ‘hroc’ (output of hROC function),
FPR	False-positive rate to be highlighted, both on the ROC curve (right plot) and classification subset (left plot). Default: 0.15.
plot.roc	If TRUE, the ROC curve is displayed. Default: TRUE.
plot.auc	If TRUE, the AUC is displayed on the plot of the bottomleft corner of the ROC curve plot. Default: FALSE.
col	Vector of length 2 with colors used for outside and inside of classification subsets, respectively. Default: c("white", "grey").
col.FPR	Color used to highlight the FPR chose. Default: "blue".
type.plotroc, lwd	Line type and width for the ROC curve. Default: type.plotroc = "s", lwd = 2.
new.window	If TRUE, graphics are displayed in a new window. Default: TRUE.
xlim	Range for x-axis on the left plot. Default: range of the marker values.
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot.

<code>cex.lab, cex.axis, cex.main</code>	The magnification to be used for labels, axis annotation and main titles, respectively, relative to the current setting of <code>cex</code> . Default: <code>cex.lab = 1.5</code> , <code>cex.axis = 1.5</code> , <code>cex.main = 1.75</code> .
<code>xlab, ylab</code>	Label for x- and y-axis on the left plot.
<code>main, main.plotroc</code>	Title for the left and the right plot, respectively.
<code>legend</code>	If TRUE, a legend with the meaning of colors is displayed on the left plot. Default: TRUE.
<code>cex.legend</code>	The magnification to be used for legend, relative to the current setting of <code>cex</code> . Default: 1.
<code>verbose</code>	If TRUE, a progress bar is displayed. Default: FALSE.
<code>...</code>	Other parameters to be passed. Not used.

**Value**

A plot of the classification regions underlying a ROC curve with the selected graphical parameters

**Examples**

```
data(HCC)
# 1. Standard ROC curve
roc_cg18384097 <- gROC(X = HCC$cg18384097, D = HCC$tumor)
plot_regions(roc_cg18384097, plot.auc = TRUE)
# 2. gROC curve
groc_cg18384097 <- gROC(X = HCC$cg18384097, D = HCC$tumor, side = "both")
plot_regions(groc_cg18384097, plot.auc = TRUE)
# 3. hROC curve with a restricted cubic splines transformation
hroc_cg18384097 <- hROC(X = HCC$cg18384097, D = HCC$tumor,
  formula.lrm = "D ~ rcs(X,8)")
plot_regions(hroc_cg18384097, plot.auc = TRUE)
```

---

predict

*Predict the classification regions for a particular specificity*

---

**Description**

This function prints the classification subsets corresponding to a particular false-positive rate FPR or to cutoff value(s) C or XL, XU introduced by the user.

**Usage**

```
## S3 method for class 'groc'
predict(object, FPR = NULL, C = NULL, XL = NULL, XU = NULL, ...)
## S3 method for class 'hroc'
predict(object, FPR = 0.15, ...)
```

**Arguments**

object	An object of class 'groc' or 'hroc'.
FPR	False-positive rate used to predict the classification region. Default: 0.15 if no cutoff value is provided by the next input parameters.
C	Cutoff value used to predict the classification region for 'groc' object with side = "right" or "left". If FPR is provided, C is not used. Default: none.
XL, XU	Cutoff values used to predict the classification region for 'groc' object with side = "both" or "both2". If FPR is provided, C is not used. Default: none.
...	Other parameters to be passed. Not used.

**Value**

A list of length 3 with the following fields:

ClassSubsets	A matrix with the classification region. Number of rows indicate the number of intervals whose union defines the classification region.
Specificity	Resulting specificity value.
Sensitivity	Resulting sensitivity value.

**Examples**

```
data(HCC)

roc <- gROC(X = HCC$cg18384097, D = HCC$tumor) # Right-sided ROC curve
predict(roc, FPR = 0.5)
groc <- gROC(X = HCC$cg18384097, D = HCC$tumor, side = "both") # gROC curve
predict(groc, FPR = 0.5)
hroc_cg18384097 <- hROC(X = HCC$cg18384097, D = HCC$tumor,
  formula.lrm = "D ~ rcs(X,8)")
predict(hroc_cg18384097, FPR = 0.5)
```

---

print

---

*Print an ROC curve object*


---

**Description**

This function prints information about an ROC curve.

**Usage**

```
## S3 method for class 'groc'
print(x, ...)
## S3 method for class 'hroc'
print(x, ...)
## S3 method for class 'multiroc'
print(x, ...)
```

**Arguments**

- x An ROC curve object from the **movieROC** package. Possible classes are: 'groc' (output of `gROC()` function), 'hroc' (output of `hROC()` function), and 'multiroc' (output of `multiROC()` function).
- ... Other parameters to be passed. Not used.

**Value**

A character vector with the details of an ROC curve object.

**Examples**

```
## See examples for gROC(), hROC(), and multiROC() function
```

# Index

## \* datasets

HCC, 5

gROC, 2, 9, 16, 18, 28

gROC\_param, 4

HCC, 5

hROC, 6, 16, 28

movieROC, 2, 8, 13

movieROC2\_densities, 11

multiROC, 9, 13, 16, 18, 28

plot, 2, 6, 13, 16

plot\_buildROC, 2, 13, 18

plot\_densities, 2, 21

plot\_densityROC, 2, 12, 22

plot\_funregions, 6, 24

plot\_regions, 2, 6, 25

predict, 2, 6, 26

print, 2, 6, 13, 27