Package 'iCOBRA'

November 12, 2026

Type Package

Title Comparison and Visualization of Ranking and Assignment Methods

Version 1.40.0

Date 2025-12-13

Description This package provides functions for calculation and visualization of performance metrics for evaluation of ranking and binary classification (assignment) methods. Various types of performance plots can be generated programmatically. The package also contains a shiny application for interactive exploration of results.

License GPL (>=2)

Collate 'AllGenerics.R' 'COBRAData.R' 'COBRAPerformance.R' 'COBRAPlot.R' 'calculate_performance.R' 'cobradata_example.R' 'cobradata_example_sval.R' 'helpers_general.R' 'plot_methods.R' 'printHead.R' 'shiny.R'

Depends R (>= 4.4.0)

Imports shiny (>= 0.9.1.9008), shinydashboard, reshape2, ggplot2 (>= 3.4.0), scales, ROCR, dplyr, DT, limma, methods, UpSetR, utils, rlang, prompter

Suggests knitr, markdown, rmarkdown, testthat

VignetteBuilder knitr

biocViews Classification, Visualization

Encoding UTF-8

RoxygenNote 7.3.3

URL https://csoneson.github.io/iCOBRA/

BugReports https://github.com/csoneson/iCOBRA/issues

git_url https://git.bioconductor.org/packages/iCOBRA

git_branch RELEASE_3_23

git_last_commit b169147

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Charlotte Soneson [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3833-2169>)

Maintainer Charlotte Soneson <charlottesoneson@gmail.com>

Contents

basemethods . . . 3
calculate_adjp . . . 4
calculate_performance . . . 5
COBRAapp . . . 7
COBRAData-class . . . 8
cobradata_example . . . 9
cobradata_example_sval . . . 10
COBRAPerformance-class . . . 11
COBRAPlot-class . . . 13
coerce . . . 15
corr . . . 16
deviation . . . 17
Extract . . . 18
facetted . . . 19
fdrnbr . . . 20
fdrnbrcurve . . . 21
fdrtpr . . . 22
fdrtprcurve . . . 23
fpc . . . 24
fpr . . . 25
fsrnbr . . . 26
fsrnbrcurve . . . 27
maxsplit . . . 28
onlyshared . . . 29
overlap . . . 30
padj . . . 31
plotcolors . . . 32
plot_corr . . . 33
plot_deviation . . . 34
plot_fdrnbrcurve . . . 35
plot_fdrtprcurve . . . 36
plot_fpc . . . 37
plot_fpr . . . 38
plot_fsrnbrcurve . . . 39
plot_overlap . . . 40
plot_roc . . . 41
plot_scatter . . . 42
plot_tpr . . . 43
plot_upset . . . 44

prepare_data_for_plot 46
pval 47
reorder_levels 48
roc 49
scatter 50
score 51
splv 52
stratiflevels 53
sval 54
tpr 55
truth 56
update_cobradata 57
update_cobraperformance 58

Index **59**

`basemethods` *Accessor function for basemethods*

Description

Accessor function to extract the methods that are represented in an `COBRAPerformance` or `COBRAPlot` object.

Usage

```
basemethods(x, ...)

## S4 method for signature 'COBRAPerformance'
basemethods(x)
```

Arguments

`x`	A `COBRAPerformance` or `COBRAPlot` object.
`...`	Additional arguments.

Value

A character vector of all methods represented in the object.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtprcurve")
basemethods(cobraperf)
```

`calculate_adjp` *Calculate adjusted p-values*

Description

Calculate adjusted p-values for methods where only nominal p-values are available in a `COBRAData` object.

Usage

```
calculate_adjp(cobradata, method = "BH")
```

Arguments

`cobradata`	A `COBRAData` object.
`method`	A character string giving the method (selected from `p.adjust.methods()`) that will be used to perform the adjustment.

Value

A `COBRAData` object, extended with the calculated adjusted p-values.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobradata_example <- calculate_adjp(cobradata_example, method = "BH")
```

`calculate_performance` *Calculate performance measures*

Description

Calculate performance measures from a given collection of p-values, adjusted p-values and scores provided in a `COBRAData` object.

Usage

```
calculate_performance(
  cobradata,
  binary_truth = NULL,
  cont_truth = NULL,
 aspects = c("fdrtpr", "fdrtprcurve", "fdrnbr", "fdrnbrcurve", "tpr", "fpr", "roc",
    "fpc", "overlap", "corr", "scatter", "deviation", "fsrnbr", "fsrnbrcurve"),
  thrs = c(0.01, 0.05, 0.1),
  svalthrs = c(0.01, 0.05, 0.1),
  splv = "none",
  maxsplit = 3,
  onlyshared = FALSE,
  thr_venn = 0.05,
  type_venn = "adjp",
  topn_venn = 100,
  rank_by_abs = TRUE,
  prefer_pval = TRUE
)
```

Arguments

`cobradata`	A `COBRAData` object.
`binary_truth`	A character string giving the name of the column of truth(cobradata) that contains the binary truth (true assignment of variables into two classes, represented by 0/1).
`cont_truth`	A character string giving the name of the column of truth(cobradata) that contains the continuous truth (a continuous value that the observations can be compared to).
`aspects`	A character vector giving the types of performance measures to calculate. Must be a subset of c("fdrtpr", "fdrtprcurve", "fdrnbr", "fdrnbrcurve", "tpr", "fpr", "roc", "fpc", "overlap", "corr", "scatter", "deviation", "fsrnbr", "fsrnbrcurve").
`thrs`	A numeric vector of adjusted p-value thresholds for which to calculate the performance measures. Affects "fdrtpr", "fdrnbr", "tpr" and "fpr".
`svalthrs`	A numeric vector of s-value thresholds for which to calculate the FSR. Affects "fsrnbr".

splv — A character string giving the name of the column of truth(cobradata) that will be used to stratify the results. The default value is "none", indicating no stratification.

maxsplit — A numeric value giving the maximal number of categories to keep in the stratification. The largest categories containing both positive and negative features will be retained. By setting this argument to 'Inf' or 'NA_integer_', all categories (as well as the order of categories) will be retained.

onlyshared — A logical, indicating whether to only consider features for which both the true assignment and a result (p-value, adjusted p-value or score) is given. If FALSE, all features contained in the truth table are used.

thr_venn — A numeric value giving the adjusted p-value threshold to use to create Venn diagrams (if type_venn is "adjp").

type_venn — Either "adjp" or "rank", indicating whether Venn diagrams should be constructed based on features with adjusted p-values below a certain threshold, or based on the same number of top-ranked features by different methods.

topn_venn — A numeric value giving the number of top-ranked features to compare between methods (if type_venn is "rank").

rank_by_abs — Whether to take the absolute value of the score before using it to rank the variables for ROC, FPC, FDR/NBR and FDR/TPR curves.

prefer_pval — Whether to preferentially rank variables by p-values or adjusted p-values rather than score for ROC and FPC calculations. From version 1.5.5, this is the default behaviour. To obtain the behaviour of previous versions, set to FALSE.

Details

Depending on the collection of observations that are available for a given method, the appropriate one will be chosen for each performance measure. For fpr, tpr, fdrtpr, fdrnbr and overlap aspects, results will only be calculated for methods where adjusted p-values are included in the COBRAData object, since these calculations make use of specific adjusted p-value cutoffs. For fdrtprcurve and fdrnbrcurve aspects, the score observations will be preferentially used, given that they are monotonically associated with the adjusted p-values (if provided). If the score is not provided, the nominal p-values will be used, given that they are monotonically associated with the adjusted p-values (if provided). In other cases, the adjusted p-values will be used also for these aspects. For roc and fpc, the score observations will be used if they are provided, otherwise p-values and, as a last instance, adjusted p-values. Finally, for the fsrnbr, corr, scatter and deviation aspects, the score observations will be used if they are provided, otherwise no results will be calculated.

Value

A COBRAPerformance object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = c("fdrtpr", "fdrtprcurve",
                                               "tpr", "roc"),
                                   thrs = c(0.01, 0.05, 0.1), splv = "none")
```

COBRAapp *Interactive shiny app to visualize results*

Description

Interactive shiny app for visualization of results. The app can be initialized with a COBRAData object. If no object is provided, truth and results are loaded into the app from text files (see the Instructions tab of the app for formatting instructions). Properly formatted text files can also be obtained using the function COBRAData_to_text.

Usage

```
COBRAapp(cobradata = NULL, autorun = FALSE, addStopButton = TRUE)
```

Arguments

cobradata	An (optional) COBRAData object. If not given, the user can load results from text files.
autorun	A logical indicating whether the app calculations should start automatically on launch, or wait for the user to press the 'Start calculation!' button.
addStopButton	Logical scalar. If TRUE (default), will add a button to stop the app (by calling shiny::stopApp).

Value

Returns (and runs) an object representing the shiny app.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
app <- COBRAapp(cobradata_example)
if (interactive()) {
  shiny::runApp(app)
}
```

COBRAData-class COBRAData *object and constructor*

Description

The COBRAData class contains slots to hold calculated p-values, adjusted p-values and general 'scores' for a set of features, as well as s-values (see Stephens (2017)). The slots can contain values from multiple methods, and each method can contribute to one or more slots. The class also contains a slot giving the 'truth' (a binary assignment and/or a continuous score) for each feature, as well as additional annotations that can be used to stratify the performance calculations.

Usage

```
COBRAData(
  pval = data.frame(),
  padj = data.frame(),
  score = data.frame(),
  sval = data.frame(),
  truth = data.frame(),
  object_to_extend = NULL
)

COBRAData_from_text(truth_file, result_files, feature_id)

COBRAData_to_text(cobradata, truth_file, result_files, feature_id)
```

Arguments

pval	A data frame with features as rows and methods as columns, containing nominal p-values. Missing values (NAs) are allowed. The row names should be feature names.
padj	A data frame with features as rows and methods as columns, containing adjusted p-values. Missing values (NAs) are allowed. The row names should be feature names.
score	A data frame with features as rows and methods as columns, containing generic scores. In case of comparison to a binary truth, larger values of the scores should correspond to 'more significant' features. Missing values (NAs) are allowed. The row names should be feature names.
sval	A data frame with features as rows and methods as columns, containing s-values (analogous to q-values, but for sign errors, see Stephens (2017)). Missing values (NAs) are allowed. The row names should be feature names.
truth	A data frame with features as rows columns containing feature annotations such as, e.g., binary and continuous truths and additional annotations that can be used to stratify the performance calculations. The row names should be feature names.

object_to_extend
: A COBRAData object to extend with the provided information.

truth_file
: A character string giving the path to a file with true labels and other feature annotations.

result_files
: A character vector giving path(s) to file(s) with results (p-values, adjusted p-values, s-values, scores) for one or more methods. The column names of these files must be of the form "method:measure", where measure is one of P, adjP, S or score, depending on what is given in the column.

feature_id
: A character string giving the name of the column in the truth and result files that encodes the feature identifier.

cobradata
: A COBRAData object

Details

If adjusted p-values are missing for some methods, for which nominal p-values are available, the adjusted p-values can be calculated using the calculate_adjp function.

The text files generated by COBRAData_to_text can be used as input to iCOBRAapp, when it is called without an input argument.

Value

COBRAData and COBRAData_from_text return a COBRAData object.

Author(s)

Charlotte Soneson

Examples

```
## Empty COBRAData object:
COBRAData()

## COBRAData object from individual data frames
set.seed(123)
pval <- data.frame(m1 = runif(100), m2 = runif(100),
                   row.names = paste0("F", 1:100))
truth <- data.frame(status = round(runif(100)),
                    row.names = paste0("F", 1:100))
cobradata <- COBRAData(pval = pval, truth = truth)
```

---

cobradata_example *Example data set with three differential gene expression methods*

Description

A data set consisting of p-values, adjusted p-values and estimated absolute log fold changes (in the 'scores' slot) obtained by three methods for differential expression analysis of RNA-seq data, applied to a small synthetic data set of 3,858 human genes. The values are stored in a `COBRAData` object together with a 'truth' data frame containing the true differential expression status for each gene as well as various additional annotations such as the true log fold change, the number of isoforms of the gene and the average expression level.

Format

A `COBRAData` object with five slots:

pval data frame with p-values for 2,399 genes, from three different methods.

padj data frame with adjusted p-values for 2,399 genes, from two different methods.

sval empty data frame

score data frame with estimated absolute log fold changes for 2,399 genes, from three different methods.

truth data frame with true differential expression status (status column), the number of isoforms (n_isoforms column), the true absolute log fold change (logFC and logFC_cat columns) and average expression level (expr and expr_cat columns) for 3,858 genes.

Value

A `COBRAData` object.

`cobradata_example_sval`

Example data set with three differential gene expression methods

---

Description

A data set consisting of p-values, adjusted p-values, s-values and estimated log fold changes (in the 'scores' slot) obtained by three methods for differential expression analysis of RNA-seq data, applied to a small synthetic data set of 3,858 human genes. The values are stored in a `COBRAData` object together with a 'truth' data frame containing the true differential expression status for each gene as well as various additional annotations such as the true log fold change, the number of isoforms of the gene and the average expression level.

Format

A `COBRAData` object with five slots:

pval data frame with p-values for 2,399 genes, from three different methods.

padj data frame with adjusted p-values for 2,399 genes, from two different methods.

sval data frame representing s-values for 2,399 genes, from two different methods. Since the purpose of this data set is only to demonstrate functionality, the s-values were obtained by copying the adjusted p-value slot, and thus these values do not represent true s-values.

score data frame with estimated log fold changes for 2,399 genes, from three different methods.

truth data frame with true differential expression status (status column), the number of isoforms (n_isoforms column), the true log fold change (logFC and logFC_cat columns) and average expression level (expr and expr_cat columns) for 3,858 genes.

Value

A COBRAData object.

COBRAPerformance-class

COBRAPerformance *object and constructor*

Description

The COBRAPerformance class holds various types of calculated performance measures. Objects from this class are typically generated from COBRAData objects by means of the function calculate_performance.

Usage

```
COBRAPerformance(
  fdrtpr = data.frame(),
  fdrtprcurve = data.frame(),
  fdrnbr = data.frame(),
  fdrnbrcurve = data.frame(),
  fsrnbr = data.frame(),
  fsrnbrcurve = data.frame(),
  tpr = data.frame(),
  fpr = data.frame(),
  splv = "",
  roc = data.frame(),
  fpc = data.frame(),
  deviation = data.frame(),
  onlyshared = NA,
  overlap = data.frame(),
  maxsplit = NA_integer_,
  corr = data.frame(),
  scatter = data.frame()
)
```

Arguments

fdrtpr	A data frame containing observed FDR and TPR values at various adjusted p-value thresholds.
fdrtprcurve	A data frame containing observed FDR and TPR values for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score).
fdrnbr	A data frame containing observed FDR and the number of features considered to be significant at various adjusted p-value thresholds.
fdrnbrcurve	A data frame containing observed FDR and number of features considered to be significant for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score).
fsrnbr	A data frame containing observed false sign rate (FSR) and the number of features considered to be significant at various s-value thresholds
fsrnbrcurve	A data frame containing observed false sign rate (FSR) and number of features considered to be significant for a (potentially large) number of cutoffs applied to the s-value
tpr	A data frame containing observed TPR values at various adjusted p-value thresholds.
fpr	A data frame containing observed FPR values at various adjusted p-value thresholds.
splv	A character string giving the name of the stratification factor, "none" if the results are not stratified.
roc	A data frame containing observed FPR and TPR values for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score), which can be used to generate a ROC curve.
fpc	A data frame containing observed numbers of false positive findings among the N top-ranked features (ranked by p-values, adjusted p-values or more general scores), for a (potentially large) number of Ns, which can be used to generate a false positive curve.
deviation	A data frame containing deviations between observed scores and true scores.
onlyshared	A logical value indicating whether only features shared between the results and the truth should be retained, or if all features present in the truth should be used.
overlap	A data frame or list of data frames with binary values indicating, for each of a number of methods and number of features, whether the method consider the feature 'positive' (significant, 1) or 'negative' (non-significant, 0). If it is a list of data frames, each list element corresponds to one level of a stratifying factor.
maxsplit	A numeric value indicating the largest number of levels to retain if the results have been stratified by an annotation.
corr	A data frame containing observed (Pearson and Spearman) correlation values between observed and true scores.
scatter	A data frame containing observed 'scores' (p-values, adjusted p-values or more general scores) and true scores, which can be used to generate scatter plots.

Value

A COBRAPerformance object.

Author(s)

Charlotte Soneson

Examples

```
## Empty COBRAPerformance object
COBRAPerformance()
```

COBRAPlot-class COBRAPlot *object and constructor*

Description

The COBRAPlot class is similar to the COBRAPerformance class in that it holds various types of calculated performance measures. However, it also contains other attributes that are necessary for plotting, such as color assignments. Several COBRAPlot objects can be generated from the same COBRAPerformance object, without having to go through the potentially time consuming task of recalculating all performance measures. Objects from this class are typically generated from a COBRAPerformance objects by means of the function prepare_data_for_plot.

Usage

```
COBRAPlot(
  fdrtpr = data.frame(),
  fdrtprcurve = data.frame(),
  fsrnbr = data.frame(),
  fsrnbrcurve = data.frame(),
  fdrnbr = data.frame(),
  corr = data.frame(),
  fdrnbrcurve = data.frame(),
  tpr = data.frame(),
  fpr = data.frame(),
  roc = data.frame(),
  scatter = data.frame(),
  onlyshared = NA,
  fpc = data.frame(),
  overlap = data.frame(),
  plotcolors = "",
  splv = "",
  deviation = data.frame(),
  maxsplit = NA_integer_,
  facetted = NA
)
```

Arguments

fdrtpr	A data frame containing observed FDR and TPR values at various adjusted p-value thresholds.
fdrtprcurve	A data frame containing observed FDR and TPR values for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score).
fsrnbr	A data frame containing observed false sign rate (FSR) and the number of features considered to be significant at various s-value thresholds
fsrnbrcurve	A data frame containing observed false sign rate (FSR) and number of features considered to be significant for a (potentially large) number of cutoffs applied to the s-value
fdrnbr	A data frame containing observed FDR and the number of features considered to be significant at various adjusted p-value thresholds.
corr	A data frame containing observed (Pearson and Spearman) correlation values between observed and true scores.
fdrnbrcurve	A data frame containing observed FDR and number of features considered to be significant for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score).
tpr	A data frame containing observed TPR values at various adjusted p-value thresholds.
fpr	A data frame containing observed FPR values at various adjusted p-value thresholds.
roc	A data frame containing observed FPR and TPR values for a (potentially large) number of cutoffs applied to a 'score' (that can be p-value, adjusted p-value or a more general score), which can be used to generate a ROC curve.
scatter	A data frame containing observed 'scores' (p-values, adjusted p-values or more general scores) and true scores, which can be used to generate scatter plots.
onlyshared	A logical value indicating whether only features shared between the results and the truth should be retained, or if all features present in the truth should be used.
fpc	A data frame containing observed numbers of false positive findings among the N top-ranked features (ranked by p-values, adjusted p-values or more general scores), for a (potentially large) number of Ns, which can be used to generate a false positive curve.
overlap	A data frame or list of data frames with binary values indicating, for each of a number of methods and number of features, whether the method consider the feature 'positive' (significant, 1) or 'negative' (non-significant, 0). If it is a list of data frames, each list element corresponds to one level of a stratifying factor.
plotcolors	A character vector giving the color for each method (or method-stratification level combination).
splv	A character string giving the name of the stratification factor, "none" if the results are not stratified.
deviation	A data frame containing deviations between observed scores and true scores.

maxsplit	A numeric value indicating the largest number of levels to retain if the results have been stratified by an annotation.
facetted	A logical indicating whether the data is prepared for a facetted plot (separating different stratification levels into different panels) or for displaying all values in one plot panel.

Value

A COBRAPlot object.

Author(s)

Charlotte Soneson

Examples

```
## Empty COBRAPlot object
cobraplot <- COBRAPlot()
```

---

coerce *Convert an object to another class*

Description

Convert object between COBRAPerformance and COBRAPlot classes.

Arguments

from	The object that is to be coerced into another class.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
cobraplot <- prepare_data_for_plot(cobraperf)

## Coerce COBRAPerformance object into COBRAPlot object
as(cobraperf, "COBRAPlot")

## Coerce COBRAPlot object into COBRAPerformance object
as(cobraplot, "COBRAPerformance")
```

corr *Accessor and replacement functions for* corr *slot*

Description

Accessor and replacement functions for the corr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
corr(x, ...)

corr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
corr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
corr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
corr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving correlation values for each method and each stratification level.

Value

The accessor function returns a data frame giving correlation values for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "corr")
head(corr(cobraperf))
```

deviation *Accessor and replacement functions for* deviation *slot*

Description

Accessor and replacement functions for the deviation slot in a COBRAPerformance or COBRAPlot object.

Usage

```
deviation(x, ...)

deviation(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
deviation(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
deviation(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
deviation(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to plots of deviations between observed and true scores for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to plots of deviations between observed and true scores for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "deviation")
head(deviation(cobraperf))
```

Extract *Subsetting* COBRAData, COBRAPerformance *or* COBRAPlot *objects*

Description

Functions to subset COBRAData, COBRAPerformance or COBRAPlot objects. COBRAData objects are subset by features (rows), while COBRAPerformance and COBRAPlot objects are subset by methods (columns). Numeric indices are not allowed, since not all slots may be arranged in the same order.

Usage

```
## S4 method for signature 'COBRAData'
x[i, j = "missing", drop = "missing"]

## S4 method for signature 'COBRAPerformance'
x[i = "missing", j, drop = "missing"]

## S4 method for signature 'COBRAPlot'
x[i = "missing", j, drop = "missing"]
```

Arguments

x	A COBRAData, COBRAPerformance or COBRAPlot object.
i	For COBRAData objects, a character vector of feature names to retain.
j	For COBRAPerformance and COBRAPlot objects, a character vector with method names to retain.
drop	not used.

Value

A subset of the original object, of the same class

Examples

```
data(cobradata_example)
cobradata_example[c("ENSG00000000457", "ENSG00000000971",
                    "ENSG00000000460"), ]
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
cobraperf[, c("voom")]
cobraplot <- prepare_data_for_plot(cobraperf)
cobraplot[, c("voom")]
```

facetted *Accessor and replacement functions for* facetted *slot*

Description

Accessor and replacement functions for the facetted slot in an COBRAPlot object.

Usage

```
facetted(x, ...)

facetted(x, ...) <- value

## S4 method for signature 'COBRAPlot'
facetted(x)

## S4 replacement method for signature 'COBRAPlot,logical'
facetted(x) <- value
```

Arguments

x	A COBRAPlot object.
...	Additional arguments.
value	A logical value, indicating whether the object is formatted for facetted plots (visualizing each stratification level in a separate panel) or not.

Value

The accessor function returns a logical value, indicating whether the object is formatted for facetted plots (visualizing each stratification level in a separate panel) or not.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
cobraplot <- prepare_data_for_plot(cobraperf)
facetted(cobraplot)
```

fdrnbr *Accessor and replacement functions for* fdrnbr *slot*

Description

Accessor and replacement functions for the fdrnbr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fdrnbr(x, ...)

fdrnbr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fdrnbr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fdrnbr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fdrnbr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information about the observed FPR and the number of features called positive for each method and each stratification level, at various adjusted p-value thresholds.

Value

The accessor function returns a data frame giving information about the observed FPR and the number of features called positive for each method and each stratification level, at various adjusted p-value thresholds.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrnbr")
head(fdrnbr(cobraperf))
```

fdrnbrcurve *Accessor and replacement functions for* fdrnbrcurve *slot*

Description

Accessor and replacement functions for the fdrnbrcurve slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fdrnbrcurve(x, ...)

fdrnbrcurve(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fdrnbrcurve(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fdrnbrcurve(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fdrnbrcurve(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to generate curves of observed FDR vs number of features called positive for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to generate curves of observed FDR vs number of features called positive for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrnbrcurve")
head(fdrnbrcurve(cobraperf))
```

fdrtpr *Accessor and replacement functions for* fdrtpr *slot*

Description

Accessor and replacement functions for the fdrtpr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fdrtpr(x, ...)

fdrtpr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fdrtpr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fdrtpr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fdrtpr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information about the observed FPR and TPR for each method and each stratification level, at various adjusted p-value thresholds.

Value

The accessor function returns a data frame giving information about the observed FPR and TPR for each method and each stratification level, at various adjusted p-value thresholds.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
head(fdrtpr(cobraperf))
```

fdrtprcurve *Accessor and replacement functions for* fdrtprcurve *slot*

Description

Accessor and replacement functions for the fdrtprcurve slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fdrtprcurve(x, ...)

fdrtprcurve(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fdrtprcurve(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fdrtprcurve(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fdrtprcurve(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to generate curves of observed FDR vs TPR for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to generate curves of observed FDR vs TPR for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtprcurve")
head(fdrtprcurve(cobraperf))
```

fpc *Accessor and replacement functions for* fpc *slot*

Description

Accessor and replacement functions for the fpc slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fpc(x, ...)

fpc(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fpc(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fpc(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fpc(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to generate false positive curves for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to generate false positive curves for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "fpc")
head(fpc(cobraperf))
```

fpr *Accessor and replacement functions for* fpr *slot*

Description

Accessor and replacement functions for the fpr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fpr(x, ...)

fpr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fpr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fpr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fpr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information about the observed FPR for each method and each stratification level, at various adjusted p-value thresholds.

Value

The accessor function returns a data frame giving information about the observed FPR for each method and each stratification level, at various adjusted p-value thresholds.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "fpr")
head(fpr(cobraperf))
```

fsrnbr *Accessor and replacement functions for* fsrnbr *slot*

Description

Accessor and replacement functions for the fsrnbr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
fsrnbr(x, ...)

fsrnbr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fsrnbr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fsrnbr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fsrnbr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information about the observed FSR and the number of features called positive for each method and each stratification level, at various s-value thresholds. If the object does not have an fsrnbr slot (older versions of the class did not have this slot), an empty data frame is returned for simplicity.

Value

The accessor function returns a data frame giving information about the observed FSR and the number of features called positive for each method and each stratification level, at various s-value thresholds.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example_sval)

cobraperf <- calculate_performance(cobradata_example_sval,
```

```
                                    cont_truth = "logFC",
                                    aspects = "fsrnbr")
head(fsrnbr(cobraperf))
```

`fsrnbrcurve` *Accessor and replacement functions for* `fsrnbrcurve` *slot*

Description

Accessor and replacement functions for the `fsrnbrcurve` slot in a `COBRAPerformance` or `COBRAPlot` object.

Usage

```
fsrnbrcurve(x, ...)

fsrnbrcurve(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
fsrnbrcurve(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
fsrnbrcurve(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
fsrnbrcurve(x) <- value
```

Arguments

`x`	A `COBRAPerformance` or `COBRAPlot` object.
`...`	Additional arguments.
`value`	A data frame giving information necessary to generate curves of observed FSR vs number of features called positive for each method and each stratification level. If the object does not have an fsrnbrcurve slot (older versions of the class did not have this slot), an empty data frame is returned for simplicity.

Value

The accessor function returns a data frame giving information necessary to generate curves of observed FSR vs number of features called positive for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example_sval)

cobraperf <- calculate_performance(cobradata_example_sval,
                                   cont_truth = "logFC",
                                   aspects = "fsrnbrcurve")
head(fsrnbrcurve(cobraperf))
```

---

maxsplit *Accessor and replacement functions for* maxsplit *slot*

Description

Accessor and replacement functions for the maxsplit slot in a COBRAPerformance or COBRAPlot object.

Usage

```
maxsplit(x, ...)

maxsplit(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
maxsplit(x)

## S4 replacement method for signature 'COBRAPerformance,numeric'
maxsplit(x) <- value

## S4 replacement method for signature 'COBRAPlot,numeric'
maxsplit(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A numeric value giving the maximal number of strata to retain.

Value

The accessor function returns a numeric value giving the maximal number of strata to retain.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr", splv = "expr_cat",
                                   maxsplit = 3)
maxsplit(cobraperf)
```

---

onlyshared *Accessor and replacement functions for* onlyshared *slot*

Description

Accessor and replacement functions for the onlyshared slot in a COBRAPerformance or COBRAPlot object.

Usage

```
onlyshared(x, ...)

onlyshared(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
onlyshared(x)

## S4 replacement method for signature 'COBRAPerformance,logical'
onlyshared(x) <- value

## S4 replacement method for signature 'COBRAPlot,logical'
onlyshared(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A logical indicating whether only features that are shared between result and truth are retained, or if all features in the truth are used.

Value

The accessor function returns a logical indicating whether only features that are shared between result and truth are retained, or if all features in the truth are used.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
head(onlyshared(cobraperf))
```

---

overlap *Accessor and replacement functions for* overlap *slot*

Description

Accessor and replacement functions for the overlap slot in a COBRAPerformance or COBRAPlot object.

Usage

```
overlap(x, ...)

overlap(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
overlap(x)

## S4 replacement method for signature 'COBRAPerformance,list_df'
overlap(x) <- value

## S4 replacement method for signature 'COBRAPlot,list_df'
overlap(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame or a list, giving information about which feature that are classified as 'positive' by each method and for each stratification level.

Value

The accessor function returns a data frame or a list, giving information about which feature that are classified as 'positive' by each method and for each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "overlap")
head(overlap(cobraperf))
```

---

padj *Accessor and replacement functions for* padj *slot*

Description

Accessor and replacement functions for the padj slot in a COBRAData object.

Usage

```
padj(x, ...)

padj(x, ...) <- value

## S4 method for signature 'COBRAData'
padj(x)

## S4 replacement method for signature 'COBRAData,data.frame'
padj(x) <- value
```

Arguments

x	A COBRAData object.
...	Additional arguments.
value	A data frame containing adjusted p-values for each feature and each method.

Value

The accessor function returns a data frame containing adjusted p-values for each feature and each method.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
head(padj(cobradata_example))
```

plotcolors *Accessor and replacement functions for* plotcolors *slot*

Description

Accessor and replacement functions for the plotcolors slot in an COBRAPlot object.

Usage

```
plotcolors(x, ...)

plotcolors(x, ...) <- value

## S4 method for signature 'COBRAPlot'
plotcolors(x)

## S4 replacement method for signature 'COBRAPlot,character'
plotcolors(x) <- value
```

Arguments

x	A COBRAPlot object.
...	Additional arguments.
value	A character vector giving the colors assigned to each of the methods (or method/stratification level combinations) represented in the COBRAPlot object.

Value

The accessor function returns a character vector giving the colors assigned to each of the methods (or method/stratification level combinations) represented in the COBRAPlot object.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr")
cobraplot <- prepare_data_for_plot(cobraperf)
plotcolors(cobraplot)
```

plot_corr *Plot correlations*

Description

Plot correlations between observations and a continuous truth value.

Usage

```
plot_corr(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(-1, 1),
  corrtype = "pearson"
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
corrtype	A character string giving the type of correlation to show. Either "pearson" or "spearman".

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "corr")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_corr(cobraplot, corrtype = "spearman")
```

`plot_deviation` *Plot deviations*

Description

Plot the deviations between observed scores and the continuous truth variable.

Usage

```
plot_deviation(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  xaxisrange = NULL,
  plottype = "boxplot",
  dojitter = TRUE,
  transf = "raw"
)
```

Arguments

`cobraplot`	A `COBRAPlot` object.
`title`	A character string giving the title of the plot.
`stripsize`	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
`titlecol`	A character string giving the color of the title.
`xaxisrange`	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
`plottype`	Either "boxplot" or "violinplot", indicating what type of plot to make.
`dojitter`	A logical indicating whether to include jittered data points or not.
`transf`	A character indicating the transformation to apply to the deviations before plotting. Must be one of "raw", "absolute" or "squared"

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "deviation")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_deviation(cobraplot)
```

---

plot_fdrnbrcurve *Plot number of significant features vs FDR*

Description

Plot the number of features considered significant vs observed false discovery rate (FDR), for given adjusted p-value thresholds and/or as curves traced out by considering all threshold values.

Usage

```
plot_fdrnbrcurve(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(0, 1),
  plottype = c("curve", "points"),
  linewidth = 1
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
plottype	A character vector giving the type of plot to construct. Can be any combination of the two elements "curve" and "points".
linewidth	The line width used for plotting

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = c("fdrnbr", "fdrnbrcurve"))
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_fdrnbrcurve(cobraplot, plottype = c("curve", "points"))
```

---

plot_fdrtprcurve *Plot TPR vs FDR*

Description

Plot observed true positive rate (TPR) vs observed false discovery rate (FDR), for given adjusted p-value thresholds and/or as curves traced out by considering all threshold values.

Usage

```
plot_fdrtprcurve(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(0, 1),
  yaxisrange = c(0, 1),
  plottype = c("curve", "points"),
  linewidth = 1
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.

pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
yaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the y-axis, respectively.
plottype	A character vector giving the type of plot to construct. Can be any combination of the two elements "curve" and "points".
linewidth	The line width used for plotting

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = c("fdrtpr", "fdrtprcurve"))
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_fdrtprcurve(cobraplot, plottype = c("curve", "points"))
```

---

plot_fpc *Plot FP curves*

Description

Plot false positive curves, indicating the number of false positives among the top-ranked N variables, for varying values of N.

Usage

```
plot_fpc(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  maxnfdc = 500,
  linewidth = 1
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
maxnfdc	A numeric value giving the largest N to consider.
linewidth	The line width used for plotting

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "fpc")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_fpc(cobraplot, maxnfdc = 750)
```

---

plot_fpr *Plot FPR*

Description

Plot observed false positive rate (FPR) for given adjusted p-value thresholds.

Usage

```
plot_fpr(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(0, 1)
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "fpr")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_fpr(cobraplot, xaxisrange = c(0, 0.25))
```

---

plot_fsrnbrcurve *Plot number of features with s-value below threshold vs FSR*

Description

Plot the number of features with an s-value below a threshold vs the observed false sign rate (FSR), for given adjusted p-value thresholds and/or as curves traced out by considering all threshold values.

Usage

```
plot_fsrnbrcurve(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(0, 1),
  plottype = c("curve", "points"),
  linewidth = 1
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
plottype	A character vector giving the type of plot to construct. Can be any combination of the two elements "curve" and "points".
linewidth	The line width used for plotting

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example_sval)
cobraperf <- calculate_performance(cobradata_example_sval,
                                   cont_truth = "logFC",
                                   aspects = c("fsrnbr", "fsrnbrcurve"))
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_fsrnbrcurve(cobraplot, plottype = c("curve", "points"))
```

---

plot_overlap *Plot Venn diagram*

Description

Plot a Venn diagram showing the overlaps among sets of significant feature for a given adjusted p-value threshold (see thr_venn argument of calculate_performance). Optionally, the truth can be included as a "perfect" method. Note that maximally five methods (including the truth, if applicable) can be compared.

Usage

```
plot_overlap(cobraplot, ...)
```

Arguments

cobraplot	A COBRAPlot object.
...	Additional arguments to limma::vennDiagram.

Value

Nothing, displays a graph

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "overlap")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_overlap(cobraplot)
```

---

plot_roc *Plot ROC curves*

Description

Plot receiver operating characteristics (ROC) curves.

Usage

```
plot_roc(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  xaxisrange = c(0, 1),
  yaxisrange = c(0, 1),
  linewidth = 1
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.

titlecol	A character string giving the color of the title.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.
yaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the y-axis, respectively.
linewidth	The line width used for plotting

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "roc")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_roc(cobraplot)
```

---

plot_scatter *Plot scatter plots*

Description

Plot scatter plots, indicating the relationship between observed values and a continuous truth.

Usage

```
plot_scatter(
  cobraplot,
  title = "",
  stripsize = 10,
  titlecol = "black",
  pointsize = 3,
  doflip = FALSE,
  dolog = FALSE
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
doflip	A logical indicating whether to flip the axes when results are stratified by an annotation. By default (doflip = FALSE), stratification levels are shown as columns and methods as rows in the plot.
dolog	A logical indicating whether to log10-transform values before plotting.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "scatter")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_scatter(cobraplot)
```

---

plot_tpr *Plot TPR*

Description

Plot observed true positive rate (TPR) for given adjusted p-value thresholds.

Usage

```
plot_tpr(
  cobraplot,
  title = "",
  stripsize = 15,
  titlecol = "black",
  pointsize = 5,
  xaxisrange = c(0, 1)
)
```

Arguments

cobraplot	A COBRAPlot object.
title	A character string giving the title of the plot.
stripsize	A numeric value giving the size of the strip text, when the results are stratified by an annotation.
titlecol	A character string giving the color of the title.
pointsize	A numeric value giving the size of the plot characters.
xaxisrange	A numeric vector with two elements, giving the lower and upper boundary of the x-axis, respectively.

Value

A ggplot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "tpr")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_tpr(cobraplot)
```

---

plot_upset *Create UpSet plots*

Description

Generate UpSet plots showing the overlaps among sets of significant feature for a given adjusted p-value threshold (see thr_venn argument of calculate_performance). Optionally, the truth can be included as a "perfect" method. Note that if the results are stratified, only one category at a time can be displayed.

Usage

```
plot_upset(
  cobraplot,
  stratum = NULL,
  nsets = NULL,
  nintersects = NULL,
  sets.bar.color = NULL,
  ...
)
```

Arguments

cobraplot	A COBRAPlot object.
stratum	If results are stratified, the category to plot results for. Can be numeric or categorical (the name of the category).
nsets	The number of methods to include. By default, it is determined automatically from the cobraplot object.
nintersects	The number of set intersections to display. By default, it is determined automatically from the cobraplot object.
sets.bar.color	The colors to use for the bars in the UpSet plot. By default, they are extracted from the plotcolors slot of the cobraplot object.
...	Additional arguments to UpSetR::upset.

Value

Nothing, displays a graph

Author(s)

Charlotte Soneson

References

Lex and Gehlenborg (2014): Points of view: Sets and intersections. Nature Methods 11, 779.

Lex et al (2014): UpSet: Visualization of intersecting sets. IEEE Transactions on Visualization and Computer Graphics 20(12), 1983-1992.

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "overlap")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_upset(cobraplot)
plot_upset(cobraplot, order.by = "freq", decreasing = TRUE)

cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "overlap",
                                   splv = "expr_cat")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
plot_upset(cobraplot, stratum = "[2.85e+00,1.45e+01)")
```

prepare_data_for_plot *Prepare data for plotting*

Description

Prepare performance data provided in a COBRAPerformance object (obtained by calculate_performance) for plotting.

Usage

```
prepare_data_for_plot(
  cobraperf,
  keepmethods = NULL,
  incloverall = TRUE,
  colorscheme = "hue_pal",
  facetted = TRUE,
  incltruth = TRUE,
  conditionalfill = TRUE
)
```

Arguments

cobraperf	A COBRAPerformance object.
keepmethods	A character vector consisting of methods to retain for plotting (these should be a subset of basemethods(cobraperf)), or NULL (indicating that all methods represented in cobraperf should be retained).
incloverall	A logical indicating whether the "overall" results should be included if the results are stratified by an annotation.
colorscheme	Either a character string giving the color palette to use to define colors for the different methods, or a character vector with colors to use. The available predefined palettes depend on the number of different methods to distinguish. The choices are: - Accent (max 8 methods) - Dark2 (max 8 methods) - Paired (max 12 methods) - Pastel1 (max 9 methods) - Pastel2 (max 8 methods) - Set1 (max 9 methods) - Set2 (max 8 methods) - Set3 (max 12 methods) - hue_pal - rainbow - heat - terrain

- topo
- cm

If the number of allowed methods is exceeded, the colorscheme defaults to hue_pal.

facetted A logical indicating whether the results should be split into subpanels when stratified by an annotation (TRUE), or kept in the same panel but shown with different colors (FALSE).

incltruth A logical indicating whether the truth should be included in Venn diagrams.

conditionalfill
A logical indicating whether the points (in FDR/TPR, FDR/NBR, FSR/NBR plots) should be filled conditional on whether they satisfy the imposed criterion (e.g., false discovery rate control at imposed threshold).

Value

A COBRAPlot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   cont_truth = "none",
                                   aspects = c("fdrtpr", "fdrtprcurve",
                                               "tpr", "roc"),
                                   thrs = c(0.01, 0.05, 0.1), splv = "none")
cobraplot <- prepare_data_for_plot(cobraperf, keepmethods = NULL,
                                   colorscheme = "Dark2")

## User-specified colors
cobraplot2 <- prepare_data_for_plot(cobraperf, keepmethods = NULL,
                                   colorscheme = c("blue", "red", "green"))
```

---

pval *Accessor and replacement functions for* pval *slot*

Description

Accessor and replacement functions for the pval slot in a COBRAData object.

Usage

```
pval(x, ...)

pval(x, ...) <- value

## S4 method for signature 'COBRAData'
pval(x)

## S4 replacement method for signature 'COBRAData,data.frame'
pval(x) <- value
```

Arguments

x	A COBRAData object.
...	Additional arguments.
value	A data frame containing p-values for each feature and each method.

Value

The accessor function returns a data frame containing p-values for each feature and each method.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
head(pval(cobradata_example))
```

---

reorder_levels *Reorder levels in COBRAPlot object*

Description

Reorder levels in COBRAPlot object to achieve desired ordering in figure legends etc. If facetted(cobraplot) is TRUE, the releveling will be applied to the "method" column. If facetted(cobraplot) is FALSE, it will be applied to the "fullmethod" column.

Usage

```
reorder_levels(cobraplot, levels)
```

Arguments

cobraplot	A COBRAPlot object
levels	A character vector giving the order of the levels. Any values not present in the COBRAPlot object will be removed. Any methods present in the COBRAPlot object but not contained in this vector will be added at the end.

Value

A COBRAPlot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example_sval)
cobraperf <- calculate_performance(cobradata_example_sval,
                                   binary_truth = "status", aspects = "fpr")
cobraplot <- prepare_data_for_plot(cobraperf, colorscheme = "Dark2",
                                   incltruth = TRUE)
cobraplot <- reorder_levels(cobraplot, c("Method2", "Method1"))
```

---

roc *Accessor and replacement functions for* roc *slot*

Description

Accessor and replacement functions for the roc slot in a COBRAPerformance or COBRAPlot object.

Usage

```
roc(x, ...)

roc(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
roc(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
roc(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
roc(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to generate ROC curves for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to generate ROC curves for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "roc")
head(roc(cobraperf))
```

---

scatter *Accessor and replacement functions for* scatter *slot*

Description

Accessor and replacement functions for the scatter slot in a COBRAPerformance or COBRAPlot object.

Usage

```
scatter(x, ...)

scatter(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
scatter(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
scatter(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
scatter(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information necessary to generate scatter plots of observed vs true values for each method and each stratification level.

Value

The accessor function returns a data frame giving information necessary to generate scatter plots of observed vs true values for each method and each stratification level.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example, cont_truth = "logFC",
                                   aspects = "scatter")
head(scatter(cobraperf))
```

---

score *Accessor and replacement functions for* score *slot*

Description

Accessor and replacement functions for the score slot in a COBRAData object.

Usage

```
score(x, ...)

score(x, ...) <- value

## S4 method for signature 'COBRAData'
score(x)

## S4 replacement method for signature 'COBRAData,data.frame'
score(x) <- value
```

Arguments

x	A COBRAData object.
...	Additional arguments.
value	A data frame containing scores for each feature and each method.

Value

The accessor function regurns a data frame containing scores for each feature and each method.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
head(score(cobradata_example))
```

---

splv *Accessor and replacement functions for* splv *slot*

Description

Accessor and replacement functions for the splv slot in a COBRAPerformance or COBRAPlot object.

Usage

```
splv(x, ...)

splv(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
splv(x)

## S4 replacement method for signature 'COBRAPerformance,character'
splv(x) <- value

## S4 replacement method for signature 'COBRAPlot,character'
splv(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A character string giving the name of a feature annotation to use for stratification.

Value

The accessor function returns a character string giving the name of a feature annotation to use for stratification.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr", splv = "expr_cat")
splv(cobraperf)
```

---

stratiflevels *Accessor function for stratification levels*

Description

Accessor function to extract the stratification levels that are represented in a `COBRAPerformance` or `COBRAPlot` object.

Usage

```
stratiflevels(x, ...)

## S4 method for signature 'COBRAPerformance'
stratiflevels(x)
```

Arguments

`x`	A `COBRAPerformance` or `COBRAPlot` object
`...`	Additional arguments

Value

A character vector of all stratification levels represented in the object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status",
                                   aspects = "fdrtpr", splv = "expr_cat",
                                   maxsplit = 4)
stratiflevels(cobraperf)
```

sval *Accessor and replacement functions for* sval *slot*

Description

Accessor and replacement functions for the sval slot in a COBRAData object.

Usage

```
sval(x, ...)

sval(x, ...) <- value

## S4 method for signature 'COBRAData'
sval(x)

## S4 replacement method for signature 'COBRAData,data.frame'
sval(x) <- value
```

Arguments

x	A COBRAData object.
...	Additional arguments.
value	A data frame containing s-values for each feature and each method. If the object does not have an s-value slot (older versions of the class did not have this slot), an empty data frame is returned for simplicity.

Value

The accessor function returns a data frame containing s-values for each feature and each method.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
head(sval(cobradata_example))
```

tpr *Accessor and replacement functions for* tpr *slot*

Description

Accessor and replacement functions for the tpr slot in a COBRAPerformance or COBRAPlot object.

Usage

```
tpr(x, ...)

tpr(x, ...) <- value

## S4 method for signature 'COBRAPerformance'
tpr(x)

## S4 replacement method for signature 'COBRAPerformance,data.frame'
tpr(x) <- value

## S4 replacement method for signature 'COBRAPlot,data.frame'
tpr(x) <- value
```

Arguments

x	A COBRAPerformance or COBRAPlot object.
...	Additional arguments.
value	A data frame giving information about the observed TPR for each method and each stratification level, at various adjusted p-value thresholds.

Value

The accessor function returns a data frame giving information about the observed TPR for each method and each stratification level, at various adjusted p-value thresholds.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
cobraperf <- calculate_performance(cobradata_example,
                                   binary_truth = "status", aspects = "tpr")
head(tpr(cobraperf))
```

truth *Accessor and replacement functions for* truth *slot*

Description

Accessor and replacement functions for the truth slot in a COBRAData object.

Usage

```
truth(x, ...)

truth(x, ...) <- value

## S4 method for signature 'COBRAData'
truth(x)

## S4 replacement method for signature 'COBRAData,data.frame'
truth(x) <- value
```

Arguments

x	A COBRAData object.
...	Additional arguments.
value	A data frame containing true assignments and/or scores for features, together with other feature annotations to use for stratification of performance calculations.

Value

The accessor function returns a data frame containing true assignments and/or scores for features, together with other feature annotations to use for stratification of performance calculations.

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example)
head(truth(cobradata_example))
```

update_cobradata *Update* COBRAData *object to the current version of the class format*

Description

Update a COBRAData object generated by a previous version of the package to the latest version.

Usage

```
update_cobradata(object, quiet = FALSE)
```

Arguments

object	A COBRAData object
quiet	Set to TRUE to disable messages listing the modifications that are applied to the object

Value

An updated COBRAData object

Author(s)

Charlotte Soneson

Examples

```
## Generate COBRAData object
set.seed(123)
pval <- data.frame(m1 = runif(100), m2 = runif(100),
                   row.names = paste0("F", 1:100))
truth <- data.frame(status = round(runif(100)),
                    row.names = paste0("F", 1:100))
cobradata <- COBRAData(pval = pval, truth = truth)

## Update object if needed
cobradata <- update_cobradata(cobradata)
```

update_cobraperformance *Update* COBRAPerformance *or* COBRAPlot *object to the current version of the class format*

Description

Update a COBRAPerformance or COBRAPlot object generated by a previous version of the package to the latest version.

Usage

```
update_cobraperformance(object, quiet = FALSE)
```

Arguments

object	A COBRAPerformance or COBRAPlot object
quiet	Set to TRUE to disable messages listing the modifications that are applied to the object

Value

An updated COBRAPerformance or COBRAPlot object

Author(s)

Charlotte Soneson

Examples

```
data(cobradata_example_sval)
cobradata <- cobradata_example_sval

cobraperf <- calculate_performance(cobradata,
                                   binary_truth = "status",
                                   aspects = "fpr")
cobraperf <- update_cobraperformance(cobraperf)
```

Index

.COBRAData *(COBRAData-class)*, 8
.COBRAPerformance *(COBRAPerformance-class)*, 11
.COBRAPlot *(COBRAPlot-class)*, 13
[*(Extract)*, 18
[,COBRAData-method *(Extract)*, 18
[,COBRAPerformance-method *(Extract)*, 18
[,COBRAPlot-method *(Extract)*, 18

basemethods, 3
basemethods,COBRAPerformance-method *(basemethods)*, 3
basemethods,COBRAPlot-method *(basemethods)*, 3

calculate_adjp, 4, *9*
calculate_performance, 5, *11*, *46*
COBRAapp, 7
COBRAData *(COBRAData-class)*, 8
COBRAData-class, 8
cobradata_example, 9
cobradata_example_sval, 10
COBRAData_from_text *(COBRAData-class)*, 8
COBRAData_to_text, *7*
COBRAData_to_text *(COBRAData-class)*, 8
COBRAPerformance *(COBRAPerformance-class)*, 11
COBRAPerformance-class, 11
COBRAPlot *(COBRAPlot-class)*, 13
COBRAPlot,COBRAPerformance-method *(coerce)*, 15
COBRAPlot-class, 13
coerce, 15
coerce, *(coerce)*, 15
coerce,COBRAPerformance,COBRAPlot-method *(coerce)*, 15
coerce,COBRAPlot,COBRAPerformance-method *(coerce)*, 15
corr, 16
corr,COBRAPerformance-method *(corr)*, 16
corr,COBRAPlot-method *(corr)*, 16
corr<- *(corr)*, 16
corr<-,COBRAPerformance,data.frame-method *(corr)*, 16
corr<-,COBRAPlot,data.frame-method *(corr)*, 16

deviation, 17
deviation,COBRAPerformance-method *(deviation)*, 17
deviation,COBRAPlot-method *(deviation)*, 17
deviation<- *(deviation)*, 17
deviation<-,COBRAPerformance,data.frame-method *(deviation)*, 17
deviation<-,COBRAPlot,data.frame-method *(deviation)*, 17

Extract, 18

facetted, 19
facetted,COBRAPlot-method *(facetted)*, 19
facetted<- *(facetted)*, 19
facetted<-,COBRAPlot,logical-method *(facetted)*, 19
fdrnbr, 20
fdrnbr,COBRAPerformance-method *(fdrnbr)*, 20
fdrnbr,COBRAPlot-method *(fdrnbr)*, 20
fdrnbr<- *(fdrnbr)*, 20
fdrnbr<-,COBRAPerformance,data.frame-method *(fdrnbr)*, 20
fdrnbr<-,COBRAPlot,data.frame-method *(fdrnbr)*, 20
fdrnbrcurve, 21
fdrnbrcurve,COBRAPerformance-method *(fdrnbrcurve)*, 21
fdrnbrcurve,COBRAPlot-method *(fdrnbrcurve)*, 21
fdrnbrcurve<- *(fdrnbrcurve)*, 21

fdrnbrcurve<-,COBRAPerformance,data.frame-method (fdrnbrcurve), 21
fdrnbrcurve<-,COBRAPlot,data.frame-method (fdrnbrcurve), 21
fdrtpr, 22
fdrtpr,COBRAPerformance-method (fdrtpr), 22
fdrtpr,COBRAPlot-method (fdrtpr), 22
fdrtpr<- (fdrtpr), 22
fdrtpr<-,COBRAPerformance,data.frame-method (fdrtpr), 22
fdrtpr<-,COBRAPlot,data.frame-method (fdrtpr), 22
fdrtprcurve, 23
fdrtprcurve,COBRAPerformance-method (fdrtprcurve), 23
fdrtprcurve,COBRAPlot-method (fdrtprcurve), 23
fdrtprcurve<- (fdrtprcurve), 23
fdrtprcurve<-,COBRAPerformance,data.frame-method (fdrtprcurve), 23
fdrtprcurve<-,COBRAPlot,data.frame-method (fdrtprcurve), 23
fpc, 24
fpc,COBRAPerformance-method (fpc), 24
fpc,COBRAPlot-method (fpc), 24
fpc<- (fpc), 24
fpc<-,COBRAPerformance,data.frame-method (fpc), 24
fpc<-,COBRAPlot,data.frame-method (fpc), 24
fpr, 25
fpr,COBRAPerformance-method (fpr), 25
fpr,COBRAPlot-method (fpr), 25
fpr<- (fpr), 25
fpr<-,COBRAPerformance,data.frame-method (fpr), 25
fpr<-,COBRAPlot,data.frame-method (fpr), 25
fsrnbr, 26
fsrnbr,COBRAPerformance-method (fsrnbr), 26
fsrnbr,COBRAPlot-method (fsrnbr), 26
fsrnbr<- (fsrnbr), 26
fsrnbr<-,COBRAPerformance,data.frame-method (fsrnbr), 26
fsrnbr<-,COBRAPlot,data.frame-method (fsrnbr), 26
fsrnbrcurve, 27
fsrnbrcurve,COBRAPerformance-method (fsrnbrcurve), 27
fsrnbrcurve,COBRAPlot-method (fsrnbrcurve), 27
fsrnbrcurve<- (fsrnbrcurve), 27
fsrnbrcurve<-,COBRAPerformance,data.frame-method (fsrnbrcurve), 27
fsrnbrcurve<-,COBRAPlot,data.frame-method (fsrnbrcurve), 27

maxsplit, 28
maxsplit,COBRAPerformance-method (maxsplit), 28
maxsplit,COBRAPlot-method (maxsplit), 28
maxsplit<- (maxsplit), 28
maxsplit<-,COBRAPerformance,numeric-method (maxsplit), 28
maxsplit<-,COBRAPlot,numeric-method (maxsplit), 28

onlyshared, 29
onlyshared,COBRAPerformance-method (onlyshared), 29
onlyshared,COBRAPlot-method (onlyshared), 29
onlyshared<- (onlyshared), 29
onlyshared<-,COBRAPerformance,logical-method (onlyshared), 29
onlyshared<-,COBRAPlot,logical-method (onlyshared), 29
overlap, 30
overlap,COBRAPerformance-method (overlap), 30
overlap,COBRAPlot-method (overlap), 30
overlap<- (overlap), 30
overlap<-,COBRAPerformance,list_df-method (overlap), 30
overlap<-,COBRAPlot,list_df-method (overlap), 30

padj, 31
padj,COBRAData-method (padj), 31
padj<- (padj), 31
padj<-,COBRAData,data.frame-method (padj), 31
plot_corr, 33
plot_deviation, 34
plot_fdrnbrcurve, 35

plot_fdrtprcurve, 36
plot_fpc, 37
plot_fpr, 38
plot_fsrnbrcurve, 39
plot_overlap, 40
plot_roc, 41
plot_scatter, 42
plot_tpr, 43
plot_upset, 44
plotcolors, 32
plotcolors,COBRAPlot-method (plotcolors), 32
plotcolors<- (plotcolors), 32
plotcolors<-,COBRAPlot,character-method (plotcolors), 32
prepare_data_for_plot, *13*, 46
pval, 47
pval,COBRAData-method (pval), 47
pval<- (pval), 47
pval<-,COBRAData,data.frame-method (pval), 47

reorder_levels, 48
roc, 49
roc,COBRAPerformance-method (roc), 49
roc,COBRAPlot-method (roc), 49
roc<- (roc), 49
roc<-,COBRAPerformance,data.frame-method (roc), 49
roc<-,COBRAPlot,data.frame-method (roc), 49

scatter, 50
scatter,COBRAPerformance-method (scatter), 50
scatter,COBRAPlot-method (scatter), 50
scatter<- (scatter), 50
scatter<-,COBRAPerformance,data.frame-method (scatter), 50
scatter<-,COBRAPlot,data.frame-method (scatter), 50
score, 51
score,COBRAData-method (score), 51
score<- (score), 51
score<-,COBRAData,data.frame-method (score), 51
splv, 52
splv,COBRAPerformance-method (splv), 52
splv,COBRAPlot-method (splv), 52
splv<- (splv), 52
splv<-,COBRAPerformance,character-method (splv), 52
splv<-,COBRAPlot,character-method (splv), 52
stratiflevels, 53
stratiflevels,COBRAPerformance-method (stratiflevels), 53
stratiflevels,COBRAPlot-method (stratiflevels), 53
sval, 54
sval,COBRAData-method (sval), 54
sval<- (sval), 54
sval<-,COBRAData,data.frame-method (sval), 54

tpr, 55
tpr,COBRAPerformance-method (tpr), 55
tpr,COBRAPlot-method (tpr), 55
tpr<- (tpr), 55
tpr<-,COBRAPerformance,data.frame-method (tpr), 55
tpr<-,COBRAPlot,data.frame-method (tpr), 55
truth, 56
truth,COBRAData-method (truth), 56
truth<- (truth), 56
truth<-,COBRAData,data.frame-method (truth), 56

update_cobradata, 57
update_cobraperformance, 58