Package 'messina'

April 16, 2019

Type Package

Title Single-gene classifiers and outlier-resistant detection of differential expression for two-group and survival problems.

Version 1.18.0

Date 2014-03-20

Description Messina is a collection of algorithms for constructing optimally robust single-gene classifiers, and for identifying differential expression in the presence of outliers or unknown sample subgroups. The methods have application in identifying lead features to develop into clinical tests (both diagnostic and prognostic), and in identifying differential expression when a fraction of samples show unusual patterns of expression.

biocViews GeneExpression, DifferentialExpression, BiomedicalInformatics, Classification, Survival

License EPL (>= 1.0)

Depends R (>= 3.1.0), survival (>= 2.37-4), methods

Imports Rcpp (>= 0.11.1), plyr (>= 1.8), ggplot2 (>= 0.9.3.1), grid (>= 3.1.0), foreach (>= 1.4.1), graphics

Suggests knitr (>= 1.5), antiProfilesData (>= 0.99.2), Biobase (>= 2.22.0), BiocStyle

Enhances doMC (>= 1.3.3)

LinkingTo Rcpp

BuildDepends roxygen2 (>= 3.1.0)

VignetteBuilder knitr

Roxygen list(wrap = TRUE)

Author Mark Pinese [aut], Mark Pinese [cre], Mark Pinese [cph]

Maintainer Mark Pinese <m.pinese@garvan.org.au>

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

git_branch RELEASE_3_8

git_last_commit e516792

git_last_commit_date 2018-10-30

Date/Publication 2019-04-15

R topics documented:

messina-package . . . 2
messina . . . 3
messinaDE . . . 5
MessinaFits-class . . . 7
MessinaParameters-class . . . 7
MessinaResult-class . . . 8
messinaSurv . . . 9
messinaTopResults . . . 11
plot,MessinaClassResult,missing-method . . . 13
plot,MessinaSurvResult,missing-method . . . 15
show,MessinaResult-method . . . 17
tcga_kirc_example . . . 18

Index **19**

`messina-package` *The Messina package for classification and outlier differential expression.*

Description

Single-gene classifiers and outlier-resistant detection of differential expression for two-group and survival problems.

Details

Messina is a collection of algorithms for constructing optimally robust single-gene classifiers, and for identifying differential expression in the presence of outliers or unknown sample subgroups. The methods have application in identifying lead features to develop into clinical tests (both diagnostic and prognostic), and in identifying differential expression when a fraction of samples show unusual patterns of expression.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

References

Pinese M, Scarlett CJ, Kench JG, et al. (2009) Messina: A Novel Analysis Tool to Identify Biologically Relevant Molecules in Disease. PLoS ONE 4(4): e5337. doi:10.1371/journal.pone.0005337

See Also

`messina`

`messinaDE`

`messinaSurv`

`messina` *Find optimal single feature classifiers*

Description

Run the Messina algorithm to find features (eg. genes) that optimally distinguish between two classes of samples, subject to minimum performance requirements.

Usage

```
messina(x, y, min_sens, min_spec, f_train = 0.9, n_boot = 50, seed = NULL,
  progress = TRUE, silent = FALSE)
```

Arguments

`x`	feature expression values, either supplied as an ExpressionSet, or as an object that can be converted to a matrix by as.matrix. In the latter case, features should be in rows and samples in columns, with feature names taken from the rows of the object.
`y`	a binary vector (TRUE/FALSE or 1/0) of class membership information for each sample in x.
`min_sens`	the minimum acceptable sensitivity that a classifier separating the two groups of y must achieve.
`min_spec`	the minimum acceptable specificity that a classifier separating the two groups of y must achieve.
`f_train`	the fraction of samples to be used in the training splits of the bootstrap rounds.
`n_boot`	the number of bootstrap rounds to use.
`seed`	an optional random seed for the analysis. If NULL, a random seed derived from the current state of the PRNG is used.
`progress`	display a progress bar tracking the computation?
`silent`	be completely silent (except for error and warning messages)?

Details

Note: If you wish to use Messina to detect differential expression, and not construct classifiers, you may find the `messinaDE` function to be a more convenient interface.

Messina constructs single-feature threshold classifiers (see below) to separate two sample groups, that are in a sense the most robust single-gene classifiers that satisfy user-supplied performance requirements. It accepts as primary input a matrix or ExpressionSet of feature data x; a vector of sample class membership y; and minimum classifier target performance values min_sens, and min_spec. Messina then examines each feature of x in turn, and attempts to build a threshold classifier that satisfies the minimum performance requirements, based on that feature. The results of this classifier training and testing are then returned in a MessinaClassResult object.

The features measured in x must be numeric and contain no missing values, but apart from that are unrestricted – common use cases are mRNA measurements and protein abundance estimates. Messina is not sensitive to the data transformation used, although for mRNA abundance measurements a log-transform or similar is suggested to aid interpretability of the results. x containing discrete values can also be examined by Messina, though if the number of possible values of the members of x is very low, the algorithm is unlikely to be very powerful.

Value

an object of class "MessinaClassResult" containing the results of the analysis.

Threshold classifiers

Messina trains single-feature threshold classifiers. These are classifiers that place unknown samples into one of two groups, based on whether the sample's measurement for a given feature is above or below a constant threshold value. They are the one-dimensional version of support vector machines (SVMs), where in this case the feature set is one-dimensional, and the 'support vector' (the threshold) is a zero-dimensional point. Threshold classifiers are defined by two properties: their threshold value, and their direction, which is the class assigned if a sample's measurement exceeds the threshold.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

References

Pinese M, Scarlett CJ, Kench JG, et al. (2009) Messina: A Novel Analysis Tool to Identify Biologically Relevant Molecules in Disease. PLoS ONE 4(4): e5337. doi:10.1371/journal.pone.0005337

See Also

MessinaClassResult-class

ExpressionSet

messinaDE

messinaSurv

Examples

```
## Load some example data
library(antiProfilesData)
data(apColonData)

x = exprs(apColonData)
y = pData(apColonData)$SubType

## Subset the data to only tumour and normal samples
sel = y %in% c("normal", "tumor")
x = x[,sel]
y = y[sel]

## Run Messina to rank probesets on their classification ability, with
## classifiers needing to meet a minimum sensitivity of 0.95, and minimum
## specificity of 0.85.
fit = messina(x, y == "tumor", min_sens = 0.95, min_spec = 0.85)

## Display the results.
fit
plot(fit)
```

`messinaDE` *Detect differential expression in the presence of outliers*

Description

Run the Messina algorithm to find differentially-expressed features (eg. genes) in the presence of outliers.

Usage

```
messinaDE(x, y, max_misattribution_rate, f_train = 0.9, n_boot = 50,
  seed = NULL, progress = TRUE, silent = FALSE)
```

Arguments

`max_misattribution_rate`	The maximum allowable sample misattribution rate, in [0, 0.5). Increasing this value will increase the algorithm's resistance to outliers, at the cost of somewhat reduced sensitivity. Note that for values >= 0.95, a conventional statistical approach to identifying differential expression (eg. t-test) will likely be more powerful than Messina. See details and the vignette for more information on selecting this parameter.
`x`	feature expression values, either supplied as an ExpressionSet, or as an object that can be converted to a matrix by as.matrix. In the latter case, features should be in rows and samples in columns, with feature names taken from the rows of the object.
`y`	a binary vector (TRUE/FALSE or 1/0) of class membership information for each sample in x.
`f_train`	the fraction of samples to be used in the training splits of the bootstrap rounds.
`n_boot`	the number of bootstrap rounds to use.
`seed`	an optional random seed for the analysis. If NULL, a random seed derived from the current state of the PRNG is used.
`progress`	display a progress bar tracking the computation?
`silent`	be completely silent (except for error and warning messages)?

Details

The Messina classification algorithm (see main page at `messina`) can be adapted to identify differentially-expressed features in a two-class setting, with tunable resistance to outliers. This convenience function simplifies the setting of parameters for this task.

Outlier differential expression

Outliers in differential expression measurements are common in many experimental contexts. They may be due to experimental errors, sample misidentification, or the presence of unknown structure (eg. disease subtypes) in what was supposed to be a homogeneous sample group. The latter two causes are particularly troublesome in clinical samples, where diagnoses can be incorrect, samples impure, and subtypes common. The effect of these outliers is to inflate within-group variance estimates, reducing the power for detecting differential expression. Messina provides a principled approach to detecting differential expression in datasets containing at most a specified level of outlier samples.

Misattribution rate

In the Messina framework, for each feature each of the two classes of samples is considered to have a typical signal level. Most samples in each class will display the level of signal that matches their class, but a small number will display a level of signal consistent with the *wrong* class. We call these samples with signal matching the wrong class 'misattributed samples'. Messina can be tuned to ignore a given rate of sample misattribution when detecting differential expression, and therefore can be smoothly adjusted to deal with varying levels of outlier contamination in an experiment.

messinaDE assumes that the probability of an outlier sample is equal in each of the two classes. There are situations where this assumption is likely incorrect: for example, in a cancer vs normal comparison, the normal samples are likely to have much more consistent expression than the highly perturbed and variable cancer samples. In these cases, the user can call the worker function `messina` directly, with min_sens and min_spec parameters set appropriately to the expected outlier rate in each class. An example of how to calculate the required parameters is given in the vignette.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

References

Pinese M, Scarlett CJ, Kench JG, et al. (2009) Messina: A Novel Analysis Tool to Identify Biologically Relevant Molecules in Disease. PLoS ONE 4(4): e5337. doi:10.1371/journal.pone.0005337

See Also

`MessinaClassResult-class`

`ExpressionSet`

`messina`

`messinaSurv`

Examples

```
## Load some example data
library(antiProfilesData)
data(apColonData)

x = exprs(apColonData)
y = pData(apColonData)$SubType

## Subset the data to only tumour and normal samples
sel = y %in% c("normal", "tumor")
x = x[,sel]
y = y[sel]

## Find differentially-expressed probesets.  Allow a sample misattribution rate of
## at most 20%.
fit = messinaDE(x, y == "tumor", max_misattribution_rate = 0.2)

## Display the results.
fit
plot(fit)
```

MessinaFits-class *The MessinaFits class*

Description

A class to store the individual messina or messinaSurv fits to a dataset.

Slots

`summary` a data frame containing summary performance measures for each feature, with features in rows, and columns:

"passed" did this feature pass the user requirements? A boolean.

"type" the type of classifier that was fit

"threshold" the threshold expression value of the classifier

"posk" the direction of the classifier

"ptrue" the fraction of bootstrap replicates in which a classifier was successfully trained.

"margin" the expression margin of the classifier

`objective_surfaces` a list of length equal to the number of features. each list entry contains a data frame of the objective function values at each threshold (cutoff) tested. Currently only populated for messinaSurv fits, with columns cutoff, objective.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

`messina`

`messinaSurv`

`MessinaResult-class`

`MessinaParameters-class`

MessinaParameters-class *The MessinaParameters class*

Description

A class to store the parameters supplied to a messina or messinaSurv

Slots

`x` a matrix of expression values supplied to the messina or messinaSurv functions. Features are in rows, samples in columns.

`y` either a vector of class membership indicators (0/1 or TRUE/FALSE), for the messina case, or a Surv object for the messinaSurv case. In either case, each entry of y should match the corresponding sample column of x.

`features` a character vector of feature ids, matching the rows of x.

`samples` a character vector of sample ids, matching the columns of x and entries of y.

`perf_requirement` a list of performance requirements. For messina results, contains named entries "min_sensitivity" and "min_specificity". For messinaSurv results, contains named entries "objective_type" and "min_objective".

`minimum_group_fraction` the size, relative to the full sample size, of the smallest subgroup that may be defined by a threshold.

`training_fraction` the fraction of samples used for training in each bootstrap round.

`num_bootstraps` the number of bootstrap iterations to perform.

`prng_seed` the PRNG seed used to initialize the PRNG before analysis.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

`messina`

`messinaSurv`

`MessinaResult-class`

MessinaResult-class *The MessinaResult class*

Description

A class to store the results of a messina or messinaSurv analysis.

Slots

`problem_type` A character string naming the variant of the messina algorithm used, either "classification" for the classification case (fit using the function messina), or "survival" for the outcome case (fit using the function messinaSurv).

`parameters` An object of class MessinaParameters, containing input data and parameters for the algorithm.

`perf_estimates` A data frame of summary performance estimates (evaluated on many out-of-bag sample draws), with one row per feature in the data matrix supplied to the fit functions (either messina or messinaSurv). For a messina fit, this contains 10 columns: Mean TPR, Mean FPR, Mean TNR, Mean FNR, Variance of TPR, Variance of FPR, Variance of TNR, Variance of FNR, Mean sensitivity, Mean specificity. For a messinaSurv fit, this contains a single column, of the mean objective value for that row's feature.

`fits` An object of class MessinaFits, containing details of the fits for each feature.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

messina

messinaSurv

MessinaParameters-class

MessinaFits-class

---

messinaSurv *Find optimal prognostic features using the Messina algorithm*

Description

Run the MessinaSurv algorithm to find features (eg. genes) that can define groups of patients with very different survival times.

Usage

```
messinaSurv(x, y, obj_min, obj_func, min_group_frac = 0.1, f_train = 0.8,
  n_boot = 50, seed = NULL, parallel = NULL, silent = FALSE)
```

Arguments

x	feature expression values, either supplied as an ExpressionSet, or as an object that can be converted to a matrix by as.matrix. In the latter case, features should be in rows and samples in columns, with feature names taken from the rows of the object.
y	a Surv object containing survival times and censoring status for each
obj_min	the minimum acceptable value of the objective metric. The metric used is specified by the parameter obj_func.
obj_func	the metric function that measures the difference in survival between patients with feature values above, and below, the threshold. Valid values are "tau", "reltau", or "coxcoef"; see details for more information.
min_group_frac	the size of the smallest sample group that is allowed to be generated by thresholding, as a fraction of the total sample. The default value of 0.1 means that no thresholds will be selected that result in a sample split yielding a group of smaller than 10 the samples. A modest value of this parameter increases the stability of the "reltau" and "coxcoef" objectives, which tend to become unstable as the number of samples in a group becomes very low; see details.
f_train	the fraction of samples to be used in the training splits of the bootstrap rounds.
n_boot	the number of bootstrap rounds to use.
seed	an optional random seed for the analysis. If NULL, the R PRNG is used as-is.
parallel	should calculations be parallelized using the doMC framework? If NULL, parallel mode is used if the doMC library is loaded, and more than one core has been registered with registerDoMC(). Note that no progress bar is displayed in parallel mode.
silent	be completely silent (except for error and warning messages)?

Details

The MessinaSurv algorithm aims to identify features for which patients with high signal and patients with low signal have very different survival outcomes. This is achieved by defininining an objective function which assigns a numerical value to how strongly the survival in two groups of patients differs, then assessing the value of this objective at different signal levels of each feature. Those features for which, at a given signal level, the objective function is consistently above a user-supplied minimum level, are selected by MessinaSurv as being single-feature survival predictors.

MessinaSurv has applications as an algorithm to identify features that are survival-related, as well as a principled method to identify threshold signal values to separate a cohort into poor- and good-prognosis subgroups. It can also be used as a feature filter, selecting and discretising survival-related features before they are input into a multivariate predictor.

Value

an object of class "MessinaSurvResult" containing the results of the analysis.

Objective functions

MessinaSurv uses the value of its objective function as a measure of the strength of the difference in survival of the two patient groups defined by the threshold. Three objective functions are currently defined:

"coxcoef" The coefficient of a Cox proportional hazards fit to the model Surv ~ I(x > T), where x is the feature signal level, and T is the threshold being tested. Range is (-inf, inf), with a no-information value of 0; positive values indicate that the subgroup defined by signal above the threshold fails sooner.

"tau" Kendall's tau for survival data, defined as (concordant + tied/2) / (concordant + discordant + tied), where concordant is the number of concordant group/survival pairs, discordant is the number of discordant group/survival pairs, and tied is the total number of tied pairs, counting both group and survival ties. Concordance is calculated expecting that samples with signal exceeding the threshold will fail sooner. Range is [0, 1], with a no-information value of 0.5. Note that the ties terms naturally penalize very high or low thresholds, and so this objective is inappropriate if somewhat unbalanced subgroups are expected to be present in the data.

"reltau" tau, normalized to remove the ties penalty. Defined as agree / (agree + disagree). Range is [0, 1], with a no-information value of 0.5. Although the ties penalty of tau is removed, and this method is thus suitable for finding unbalanced subgroups, it is now unstable at extreme threshold values (as in these cases, agree + disagree -> 0). For this reason, min_group_frac must be set to a modest value when using "reltau", to preserve stability.

Methods "coxcoef" and "reltau" show instability for very high and low threshold values, and so should be used with an appropriate value of min_group_frac for stable fits. Method "tau" is stable to extreme threshold values, and therefore will tolerate min_group_frac = 0, however note that "tau" naturally penalizes small subgroups, and is therefore a poor choice unless you wish to find approximately equal-sized subgroups.

Minimum group fraction

The parameter min_group_frac limits the size of the smallest subgroups that messinaSurv can select. As the groups become smaller, the "reltau" and "coxcoef" objective functions become unstable, and can generate spurious results. These are seen on the diagnostics produced by the messina plot functions as very high objective values at very low and high threshold values. To control these results, set min_group_frac to a high enough value that the objective functions reliably fit.

Generally, max(0.1, 10/N), where N is the total number of patients, is sufficient. Keep in mind that setting this parameter too high will limit messinaSurv's ability to identify small subsets of patients with dramatically different survival from the rest: the smallest subset that will be reliably identified is min_group_frac of patients.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

MessinaSurvResult-class

ExpressionSet

messina

messinaDE

Examples

```
## Load a subset of the TCGA renal clear cell carcinoma data
## as an example.
data(tcga_kirc_example)

## Run the messinaSurv analysis on these data.  Use a tau
## objective, with a minimum performance of 0.6.  Note that
## messinaSurv analyses are very computationally-intensive,
## so in actual use multicore use with doMC and parallel = TRUE
## is strongly recommended.
fit = messinaSurv(kirc.exprs, kirc.surv, obj_func = "tau", obj_min = 0.6)

fit
plot(fit)
```

---

messinaTopResults *Display a summary of the top results from a Messina analysis*

Description

Sorts the summary results of a Messina analysis in decreasing order of classifier margin, and displays the top n. The full sorted data.frame is invisibly returned. If n == 0, displays nothing, but still invisibly returns the full data.frame.

Usage

```
messinaTopResults(result, n = 10)
```

Arguments

result	the result returned by a call to messina, messinaDE, or messinaSurv.
n	the maximum number of top hits to display (default 10). If zero, no results are displayed, but the full data.frame of results is still returned.

Details

The displayed data.frame has the following columns. Users are encouraged to consult the vignette for a tutorial on how to interpret these results for classification and gene expression tasks.

"Rowname" The feature ID. If the data x supplied to the messina function was an ExpressionSet object, the featureName of the relevant feature of x. Otherwise, if x was a matrix with row names, the row name of the corresponding entry in the matrix, or if x was a matrix without row names, F<n>, where <n> is the row number of the corresponding row of x.

"Passed Requirements" Logical: when its performance was assessed by bootstrapping, did this feature pass the user-supplied performance requirements on out-of-bag data?

"Classifier Type" A string indicating the type of single-gene classifier that Messina fit to this feature. Valid values are given below, but for most users only the Threshold type is relevant, with the others being only of diagnostic relevance.

- **"Threshold"** A threshold classifer: samples with feature signal at or below the threshold are in one group; samples with feature signal above the threshold are in the other. This is the main result of interest in a Messina analysis, and other classifier types are more of diagnostic interest.
- **"Random"** A random (also known as Zero-Rule) classifier. In this case, the feature did not contain sufficient information to construct a good classifier, but the performance requirements were so lenient that simple guessing of an unknown sample's class based on marginal probabilities was enough to satisfy them. The presence of these 'fits' in the top results is indicative of too lenient performance requirements, or a dataset with no predictive value for the classes of interest (at least for single-feature threshold classifiers).
- **"OneClass"** All samples are always called as a single class, and this strategy is sufficient to satisfy the supplied performance requirements. Similar to the "Random" type, the presence of these results are an indicator of too lenient performance requirements.
- **"NA"** The feature was not successfully fit. Seen as an indicator of failed fitting in MessinaSurv analyses only, where the Random and OneClass defaults are not applicable.

"Threshold Value" For a Threshold classifier, the value of the optimal threshold selected by the algorithm. This is the value to use as a cutoff in separating the samples into two classes, either "Group 0" and "Group 1", or "Long surviors" and "Short survivors".

"Direction" The direction of the threshold classifier. Can take values of either -1 or 1. If -1, samples with expression above the threshold are in group 1 (/TRUE), or have shorter survival times. If 1, samples with expression value above the threshold are in group 0 (/FALSE), and have longer survival times.

"Margin" The value of the threshold classifier's margin. This is the primary measure of fit strength in a Messina analysis: a higher margin indicates stronger robustness to noise and experimental variations in a classification context, and a higher likelihood of differential expression in a gene expression context.

Value

(invisible) the full table of hits, as a data.frame sorted in order of decreasing margin.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

messina

messinaDE

messinaSurv

MessinaClassResult-class

Examples

```
## Load some example data
library(antiProfilesData)
data(apColonData)

x = exprs(apColonData)
y = pData(apColonData)$SubType

## Subset the data to only tumour and normal samples
sel = y %in% c("normal", "tumor")
x = x[,sel]
y = y[sel]

## Find differentially-expressed probesets.  Allow a sample misattribution rate of
## at most 20%.
fit = messina(x, y == "tumor", min_sens = 0.95, min_spec = 0.85)

## Print the 20 probesets with the strongest evidence for differential expression
## between tumour and normal.  Save the full table of summary results for later use.
summary_table = messinaTopResults(fit, 20)

## Access the top five probesets in the table
summary_table[1:5,]

## Examine the summary results for particular probes
summary_table[c("204719_at", "207502_at"),]
```

---

plot,MessinaClassResult,missing-method

Plot the results of a Messina analysis on a classification / differential expression problem.

Description

Produces a separate plot for each supplied feature index (either as an index into the expression data x as-supplied, or as an index into the features sorted by Messina margin, depending on the value of sort_features), showing sample expression levels, group membership, threshold value, and margin locations. Two different types of plots can be produced. See the vignette for examples.

Usage

```
## S4 method for signature 'MessinaClassResult,missing'
plot(x, y, ...)
```

Arguments

x	the result of a Messina analysis, as returned by functions messina or messinaDE.
...	additional options to control the plot:
	indices a vector of indices of features to plot. If sort_features == FALSE, the indices are into the unsorted features, as originally supplied in x supplied to messina or messinaDE. If sort_features == TRUE, features are first sorted in order of decreasing margin, and then the indices in this parameter are plotted. For example, if indices == 2 and sort_features == FALSE, the second feature in x will be plotted. However, if sort_features == TRUE, the feature with the second best classifier margin will be plotted.
	sort_features a boolean indicating whether to sort features by decreasing margin size before selecting from indices. This affects the interpretation of the parameter 'indices'; for more details see the description of that parameter.
	plot_type a string giving the type of plot to produce, either "point" or "bar". "bar" is the default, and shows expression levels as horizontal bars. Although this representation is familiar, it can be misleading in the case of log-transformed data. In that case, the "point" plot type is preferable.
y	the y coordinates of points in the plot, *optional* if x is an appropriate structure.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

MessinaClassResult-class

messina

messinaDE

Examples

```
## Load some example data
library(antiProfilesData)
data(apColonData)

x = exprs(apColonData)
y = pData(apColonData)$SubType

## Subset the data to only tumour and normal samples
sel = y %in% c("normal", "tumor")
x = x[,sel]
y = y[sel]

## Run Messina to rank probesets on their classification ability, with
## classifiers needing to meet a minimum sensitivity of 0.95, and minimum
## specificity of 0.85.
fit = messina(x, y == "tumor", min_sens = 0.95, min_spec = 0.85)

## Make bar plots of the five best fits
plot(fit, indices = 1:5, sort_features = TRUE, plot_type = "bar")
```

```
## Make a point plot of the fit to the 10th feature
plot(fit, indices = 10, sort_features = FALSE, plot_type = "point")
```

`plot,MessinaSurvResult,missing-method`
Plot the results of a Messina analysis on a survival problem.

Description

Plots diagnostic and performance information for fits in a MessinaSurvResult object, as returned by `messinaSurv`.

Usage

```
## S4 method for signature 'MessinaSurvResult,missing'
plot(x, y, ...)
```

Arguments

`x`	the result of a Messina survival analysis, as returned by `messinaSurv`.
`...`	additional options to control the plot:
	`indices` a vector of indices of features to plot. If sort_features == FALSE, the indices are into the unsorted features, as originally supplied in x supplied to messinaSurv. If sort_features == TRUE, features are first sorted in order of decreasing margin, and then the indices in this parameter are plotted. For example, if indices == 2 and sort_features == FALSE, the second feature in x will be plotted. However, if sort_features == TRUE, the feature with the second best classifier margin will be plotted.
	`sort_features` a boolean indicating whether to sort features by decreasing margin size before selecting from indices. This affects the interpretation of the parameter 'indices'; for more details see the description of that parameter.
	`bootstrap_type` a string giving the type of bootstrap error band to produce on the survival prediction plots. Can take three values: "none", "stdev", and "ci". "none", the default, plots no error bands. "stdev" performs multiple rounds of Kaplan-Meier curve estimation on bootstrap samples, and plots prediction bands corresponding to +/- 1 bootstrap standard deviation from the mean. "ci" performs bootstrapping as per "stdev", and plots prediction bands corresponding to the bootstrap_ci intervals.
	`bootstrap_ci` a value in (0.5, 1) giving the confidence interval for bootstrap_type == "ci". Ignored otherwise. Default 0.9 for 90% confidence intervals.
	`nboot` the number of bootstrap iterations to perform for calculations. Set to a reasonable default taking into account bootstrap_type and bootstrap_ci, so ordinarily does not need to be specified by the user.
	`parallel` a logical indicating whether multiprocessing using doMC should be used for the bootstrap calculations. If NULL, multiprocessing will be used if doMC is loaded and more than one parallel worker is registered.
`y`	the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

Details

For each feature index given by indices, produces four plots:

"Objective function" A plot of the value of the objective function over all possible thresholds. Each sample is represented by a point on the objective function trace. The selected threshold, if any, is shown by a solid vertical line, and the margins by dotted vertical lines on either side of it. The minimum values of the objective function specified by the user are shown as horizontal dotted lines. This plot is useful for assessing fit stability, particularly for the "coxcoef" and "reltau" objective functions, which can be unstable at low or high threshold values. See messinaSurv for details.

"Separation performance at threshold" This Kaplan-Meier plot shows two traces, showing the outcomes of the two subgroups in the cohort defined by whether the plotted feature is above or below the threshold. Optionally (if bootstrap_type != "none"), the KM traces will be surrounded by shaded regions that represent either +/- 1 SD (bootstrap_type == "stdev") or a bootstrap_ci confidence interval (bootstrap_type == "ci").

"Separation performance at lower margin" This plot is identical to the above, except that the performance when the lower margin is used to separate the sample groups is shown.

"Separation performance at lower margin" This plot is identical to the above, except that the performance when the upper margin is used to separate the sample groups is shown. These last two plots give an indication of the robustness of the MessinaSurv fit at its extremes.

The Kaplan-Meier plots may optionally display bootstrap bands, if bootstrap_type != "none". Note that the calculation of bootstrap bands is computationally-intensive, and this function will by default use multiprocessing to speed calculations if doMC is loaded and more than one core registered for use. For examples of the plots and their interpretation, see the vignette.

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

MessinaSurvResult-class

messinaSurv

Examples

```
## Load a subset of the TCGA renal clear cell carcinoma data
## as an example.
data(tcga_kirc_example)

## Run the messinaSurv analysis on these data.  Use a tau
## objective, with a minimum performance of 0.6.  Note that
## messinaSurv analyses are very computationally-intensive,
## so multicore use with doMC loaded and parallel = TRUE is
## strongly recommended.  In this example we use a single
## core by default.
fit = messinaSurv(kirc.exprs, kirc.surv, obj_func = "tau", obj_min = 0.6)

## Plot the three best features found by Messina
plot(fit, indices = 1:3)

## Plot the best feature found by Messina, with 90% confidence bands.
```

```
## Note that the bootstrap iterations can be slow, so it is
## recommended that multiple cores are used, with doMC loaded
## and parallel = TRUE.
plot(fit, indices = 1, bootstrap_type = "ci", bootstrap_ci = 0.9)

## Plot the Messina fit of the 10th feature in the dataset, with
## +/- 1 standard deviation bands.
plot(fit, indices = 10, sort_features = FALSE, bootstrap_type = "stdev")
```

---

show,MessinaResult-method

Generic show methods for Messina objects.

Description

Generic show methods for Messina objects.

Usage

```
## S4 method for signature 'MessinaResult'
show(object)

## S4 method for signature 'MessinaParameters'
show(object)

## S4 method for signature 'MessinaFits'
show(object)
```

Arguments

object	Any R object

Details

For details of the objects and their generation, see the relevant class documentation, and entries for the main functions messina, messinaDE, and messinaSurv,

Author(s)

Mark Pinese <m.pinese@garvan.org.au>

See Also

MessinaResult-class

MessinaParameters-class

MessinaFits-class

messina

messinaDE

messinaSurv

`tcga_kirc_example` *Example TCGA KIRC RNAseq expression and survival data*

Description

A small subset of the TCGA KIRC (kidney renal clear cell carcinoma) expression and survival data, for use as an example for messinaSurv.

Usage

```
tcga_kirc_example
```

Format

a `matrix` of RNAseq (TCGA platform "illuminahiseq_rnaseqv2") expression estimates kirc.exprs, with genes in rows and patients in columns; and a `Surv` object kirc.surv, giving patient survival times and status.

Author(s)

Mark Pinese, 20 March 2014.

Source

TCGA, downloaded on 16 Jan 2014, and only a small random subset of genes retained, to reduce size.

Index

∗Topic **package**
    messina-package, 2

ExpressionSet, *4*, *6*, *11*

kirc.exprs *(*tcga_kirc_example*)*, 18
kirc.surv *(*tcga_kirc_example*)*, 18

messina, *2*, 3, *5–9*, *11*, *13*, *14*, *17*
messina-package, 2
MessinaClassResult-class
        *(*MessinaResult-class*)*, 8
messinaDE, *2–4*, 5, *11*, *13*, *14*, *17*
MessinaFits-class, 7
MessinaParameters-class, 7
MessinaResult-class, 8
messinaSurv, *2*, *4*, *6–9*, 9, *13*, *15–17*
MessinaSurvResult-class
        *(*MessinaResult-class*)*, 8
messinaTopResults, 11

plot,MessinaClassResult,missing-method,
        13
plot,MessinaClassResult-method
        *(*plot,MessinaClassResult,missing-method*)*,
        13
plot,MessinaSurvResult,missing-method,
        15
plot,MessinaSurvResult-method
        *(*plot,MessinaSurvResult,missing-method*)*,
        15

show,MessinaFits-method
        *(*show,MessinaResult-method*)*, 17
show,MessinaParameters-method
        *(*show,MessinaResult-method*)*, 17
show,MessinaResult-method, 17

tcga_kirc_example, 18