

Package ‘enveomics.R’

February 5, 2019

Version 1.4.4

Title Various Utilities for Microbial Genomics and Metagenomics

Description A collection of functions for microbial ecology and other applications of genomics and metagenomics. Companion package for the Enveomics Collection (Rodriguez-R, L.M. and Konstantinidis, K.T., 2016 <DOI:10.7287/peerj.preprints.1900v1>).

Author Luis M. Rodriguez-R [aut, cre]

Maintainer Luis M. Rodriguez-R <lmrodriguezr@gmail.com>

URL <http://enve-omics.ce.gatech.edu/enveomics>

Depends R (>= 2.9), stats, methods, parallel, fitdistrplus, sn, investr

Suggests tools, vegan, ape, picante, gplots, optparse

License Artistic-2.0

LazyLoad yes

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-05 18:53:20 UTC

R topics documented:

enveomics.R-package	3
enve.barplot	3
enve.cliopts	4
enve.col.alpha	5
enve.col2alpha	6
enve.df2dist	6
enve.df2dist.group	7
enve.df2dist.list	8
enve.growthcurve	9
enve.GrowthCurve-class	10

enve.prune.dist	11
enve.recplot	12
enve.recplot2	14
enve.RecPlot2-class	15
enve.recplot2.ANlr	16
enve.recplot2.changeCutoff	16
enve.recplot2.compareIdentities	17
enve.recplot2.coordinates	17
enve.recplot2.corePeak	18
enve.recplot2.extractWindows	19
enve.recplot2.findPeaks	20
enve.recplot2.findPeaks.em	20
enve.recplot2.findPeaks.emauto	21
enve.recplot2.findPeaks.mower	22
enve.recplot2.findPeaks.__emauto_one	23
enve.recplot2.findPeaks.__em_e	24
enve.recplot2.findPeaks.__em_m	25
enve.recplot2.findPeaks.__mower	25
enve.recplot2.findPeaks.__mow_one	26
enve.RecPlot2.Peak-class	27
enve.recplot2.seqdepth	28
enve.recplot2.windowDepthThreshold	28
enve.recplot2.__counts	29
enve.recplot2.__peakHist	30
enve.recplot2.__whichClosestPeak	30
enve.tribs	31
enve.TRIBS-class	32
enve.TRIBS.merge	33
enve.tribs.test	33
enve.TRIBStest-class	34
enve.truncate	35
enve.__prune.iter	35
enve.__prune.reduce	36
enve.__tribs	36
growth.curves	37
phyla.counts	37
plot.enve.GrowthCurve	38
plot.enve.RecPlot2	39
plot.enve.TRIBS	40
plot.enve.TRIBStest	41
summary.enve.GrowthCurve	42
summary.enve.TRIBS	43
summary.enve.TRIBStest	43
\$-methods	44

enveomics.R-package *Various Utilities for Microbial Genomics and Metagenomics*

Description

A collection of functions for microbial ecology and other applications of genomics and metagenomics. Companion package for the Enveomics Collection (Rodriguez-R, L.M. and Konstantinidis, K.T., 2016 <DOI:10.7287/peerj.preprints.1900v1>).

Author(s)

Luis M. Rodriguez-R <lmrodriguezr@gmail.com> [aut, cre]

enve.barplot *enve barplot*

Description

Creates nice barplots from tab-delimited tables

Usage

```
enve.barplot(x, sizes, top = 25, colors.per.group = 9, bars.width = 4,
  legend.ncol = 1, other.col = "#000000", add.trend = FALSE,
  organic.trend = FALSE, sort.by = median, min.report = 101,
  order = NULL, col, ...)
```

Arguments

x	Can be either the input data or the path to the file containing the table. If it contains the data, it must be a data frame or an object coercible to data frame. If it is a path, it must point to a tab-delimited file containing header (first row) and row names (first column).
sizes	A numeric vector containing the real size of the samples (columns) in the same order of the input table. If set, the values are assumed to be 100%, otherwise the sum of the columns is used.
top	Maximum number of categories to display. Any additional categories will be listed as "Others".
colors.per.group	Number of categories in the first two saturation groups of colors. The third group contains the remaining categories if needed.
bars.width	Width of the barplot with respect to the legend.
legend.ncol	Number of columns in the legend.
other.col	Color of the "Others" category.

add.trend	Controls if semi-transparent areas are to be plotted between the bars to connect the regions (trend regions).
organic.trend	Controls if the trend regions are to be smoothed (curves). By default, trend regions have straight edges. If TRUE, forces add.trend=TRUE.
sort.by	Any function that takes a numeric vector and returns a numeric scalar. This function is applied to each row, and the resulting values are used to sort the rows (decreasingly). Good options include: sd, min, max, mean, median.
min.report	Minimum percentage to report the value in the plot. Any value above 100 indicates that no values are to be reported.
order	Controls how the rows should be ordered. If NULL (default), sort.by is applied per row and the results are sorted decreasingly. If NA, no sorting is performed, i.e., the original order is respected. If a vector is provided, it is assumed to be the custom order to be used (either by numeric index or by row names).
col	Colors to use. If provided, overrides the variables 'top' and 'colors.per.group', but 'other.col' is still used if the vector is insufficient for all the rows. An additional palette is available with col='coto' (contributed by Luis (Coto) Orellana).
...	Any additional parameters to be passed to 'barplot'.

Author(s)

Luis M. Rodriguez-R [aut, cre]

Examples

```
# Load data
data("phyla.counts", package="enveomics.R", envir=environment())
# Create a barplot sorted by variance with organic trends
enve.barplot(phyla.counts, # Counts of phyla in four sites
  sizes=c(250,100,75,200), # Total sizes of the datasets of each site
  bars.width=2, # Decrease from default, so the names are fully displayed
  organic.trend=TRUE, # Nice curvy background
  sort.by=var # Sort by variance across sites
)
```

enve.cliopts

enve cliopts

Description

Generates nicely formatted command-line interfaces for functions (`_closures_` only).

Usage

```
enve.cliopts(fx, rd_file, positional_arguments, usage, mandatory = c(),
  vectorize = c(), ignore = c(), number = c(), defaults = list(),
  o_desc = list(), p_desc = "")
```

Arguments

<code>fx</code>	Function for which the interface should be generated.
<code>rd_file</code>	(Optional) .Rd file with the standard documentation of the function.
<code>positional_arguments</code>	(Optional) Number of <code>_positional_</code> arguments passed to <code>parse_args</code> (package:optparse).
<code>usage</code>	(Optional) Usage passed to <code>OptionParser</code> (package:optparse).
<code>mandatory</code>	Mandatory arguments.
<code>vectorize</code>	Arguments of the function to vectorize (comma-delimited). If numeric, use also <code>'number'</code> .
<code>ignore</code>	Arguments of the function to ignore.
<code>number</code>	Force these arguments as numerics. Useful for numeric vectors (see <code>'vectorize'</code>) or arguments with no defaults.
<code>defaults</code>	Defaults to use instead of the ones provided by the formals.
<code>o_desc</code>	Descriptions of the options. Help from <code>'rd'</code> is ignored for arguments present in this list.
<code>p_desc</code>	Description of the function. Help from <code>'rd'</code> is ignored for the function description unless this value is an empty string.

Value

Returns a `'list'` with keys: `'options'`, a named list with the values for the function's arguments; and `'args'`, a vector with zero or more strings containing the positional arguments.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.col.alpha

enve col alpha

Description

Modify alpha in a color (or vector of colors).

Usage

```
enve.col.alpha(col, alpha = 1/2)
```

Arguments

<code>col</code>	Color or vector of colors. It can be any value supported by <code>'col2rgb'</code> , such as <code>'darkred'</code> or <code>'#009988'</code> .
<code>alpha</code>	Alpha value to add to the color, from 0 to 1.

Value

Returns a color or a vector of colors in hex notation including alpha.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.col2alpha *enve col2alpha*

Description

Takes a vector of colors and sets the alpha.

Usage

```
enve.col2alpha(x, alpha)
```

Arguments

x A vector of any value base colors.
alpha Alpha level to set (in the 0-1 range).

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.df2dist *enve df2dist*

Description

Transform a dataframe (or coercible object, like a table) into a 'dist' object.

Usage

```
enve.df2dist(x, obj1.index = 1, obj2.index = 2, dist.index = 3,  
          default.d = NA, max.sim = 0)
```

Arguments

<code>x</code>	A table (or coercible object) with at least three columns: (1) ID of the object 1, (2) ID of the object 2, and (3) distance between the two objects.
<code>obj1.index</code>	Index of the column containing the ID of the object 1.
<code>obj2.index</code>	Index of the column containing the ID of the object 2.
<code>dist.index</code>	Index of the column containing the distance.
<code>default.d</code>	Default value (for missing values)
<code>max.sim</code>	If not-zero, assumes that the values are similarity (not distance) and this is the maximum similarity (corresponding to distance 0). Applies transformation: $distance = (max.sim - values)/max.sim$.

Value

Returns a 'dist' object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

`enve.df2dist.group` *enve df2dist group*

Description

Transform a dataframe (or coercible object, like a table) into a 'dist' object, where there are 1 or more distances between each pair of objects.

Usage

```
enve.df2dist.group(x, obj1.index = 1, obj2.index = 2, dist.index = 3,
  summary = median, empty.rm = TRUE)
```

Arguments

<code>x</code>	A dataframe (or coercible object) with at least three columns: (1) ID of the object 1, (2) ID of the object 2, and (3) distance between the two objects.
<code>obj1.index</code>	Index of the column containing the ID of the object 1.
<code>obj2.index</code>	Index of the column containing the ID of the object 2.
<code>dist.index</code>	Index of the column containing the distance.
<code>summary</code>	Function summarizing the different distances between the two objects.
<code>empty.rm</code>	Remove rows with empty or NA groups

Value

Returns a 'dist' object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.df2dist.list *enve df2dist list*

Description

Transform a dataframe (or coercible object, like a table) into a ‘dist’ object.

Usage

```
enve.df2dist.list(x, groups, obj1.index = 1, obj2.index = 2,  
  dist.index = 3, empty.rm = TRUE, ...)
```

Arguments

x	A dataframe (or coercible object) with at least three columns: (1) ID of the object 1, (2) ID of the object 2, and (3) distance between the two objects.
groups	Named array where the IDs correspond to the object IDs, and the values correspond to the group.
obj1.index	Index of the column containing the ID of the object 1.
obj2.index	Index of the column containing the ID of the object 2.
dist.index	Index of the column containing the distance.
empty.rm	Remove incomplete matrices
...	Any other parameters supported by ‘enve.df2dist.group’.

Value

Returns a ‘list’ of ‘dist’ object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.growthcurve *enve growthcurve*

Description

Calculates growth curves using the logistic growth function.

Usage

```
enve.growthcurve(x, times = 1:nrow(x), triplicates = FALSE, design,
  new.times = seq(min(times), max(times), length.out = length(times) *
    10), level = 0.95, interval = c("confidence", "prediction"),
  plot = TRUE, FUN = function(t, K, r, P0) K * P0 * exp(r *
    t)/(K + P0 * (exp(r * t) - 1)), nls.opt = list(), ...)
```

Arguments

x	Data frame (or coercible) containing the observed growth data (e.g., O.D. values). Each column is an independent growth curve and each row is a time point. NA's are allowed.
times	Vector with the times at which each row was taken. By default, all rows are assumed to be part of constantly periodic measurements.
triplicates	If TRUE, the columns are assumed to be sorted by sample with three replicates by sample. It requires a number of columns multiple of 3.
design	Experimental design of the data. An 'array' of mode list with sample names as index and the list of column names in each sample as the values. By default, each column is assumed to be an independent sample if 'triplicates' is FALSE, or every three columns are assumed to be a sample if 'triplicates' is TRUE. In the latter case, samples are simply numbered.
new.times	Values of time for the fitted curve.
level	Confidence (or prediction) interval in the fitted curve.
interval	Type of interval to be calculated for the fitted curve.
plot	Should the growth curve be plotted?
FUN	Function to fit. By default: logistic growth with parameters 'K': carrying capacity, 'r': intrinsic growth rate, and 'P0': Initial population.
nls.opt	Any additional options passed to 'nls'.
...	Any additional parameters to be passed to 'plot.enve.GrowthCurve'.

Value

Returns an 'enve.GrowthCurve' object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

Examples

```
# Load data
data("growth.curves", package="enveomics.R", envir=environment())
# Generate growth curves with different colors
g <- enve.growthcurve(growth.curves[,-1], growth.curves[,1], triplicates=TRUE)
# Generate black-and-white growth curves with different symbols
plot(g, pch=15:17, col="black", band.density=45, band.angle=c(-45,45,0))
```

enve.GrowthCurve-class

enve.GrowthCurve S4 class

Description

Enve-omics representation of fitted growth curves.

Objects from the Class

Objects can be created by calls of the form `new(enve.GrowthCurve ...)`

Slots

design: (array) Experimental design of the experiment.

models: (list) Fitted growth curve models.

predict: (list) Fitted growth curve values.

call: (call) Call producing this object.

Methods

`$ signature(x = "enve.GrowthCurve"): ...`

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.prune.dist	<i>enve prune dist</i>
-----------------	------------------------

Description

Automatically prunes a tree, to keep representatives of each clade.

Usage

```
enve.prune.dist(t, dist.quantile = 0.25, min_dist, quiet = FALSE,  
               max_iters = 100, min_nodes_random = 40000, random_nodes_frx = 1)
```

Arguments

t	A 'phylo' object or a path to the Newick file.
dist.quantile	The quantile of edge lengths.
min_dist	The minimum distance to allow between two tips. If not set, dist.quantile is used instead to calculate it.
quiet	Boolean indicating if the function must run without output.
max_iters	Maximum number of iterations.
min_nodes_random	Minimum number of nodes to trigger "tip-pairs" nodes sampling. This sampling is less reproducible and more computationally expensive, but it's the only solution if the cophenetic matrix exceeds $2^{31}-1$ entries; above that, it cannot be represented in R.
random_nodes_frx	Fraction of the nodes to be sampled if more than 'min_nodes_random'.

Value

Returns a pruned phylo object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot *enve recplot*

Description

Produces recruitment plots provided that BlastTab.catsbj.pl has been previously executed. Requires the gplots library.

Usage

```
enve.recplot(prefix, id.min = NULL, id.max = NULL, id.binsize = NULL,
             id.splines = 0, id.metric = "id", id.summary = "sum", pos.min = 1,
             pos.max = NULL, pos.binsize = 1000, pos.splines = 0, rec.col1 = "white",
             rec.col2 = "black", main = NULL, contig.col = grey(0.85),
             ret.recplot = FALSE, ret.hist = FALSE, ret.mode = FALSE,
             id.cutoff = NULL, verbose = TRUE, ...)
```

Arguments

prefix	Path to the prefix of the BlastTab.catsbj.pl output files. At least the files .rec and .lim must exist with this prefix.
id.min	Minimum identity to be considered. By default, the minimum detected identity. This value is a percentage.
id.max	Maximum identity to be considered. By default, 100.
id.binsize	Size of the identity bins (vertical histograms). By default, 0.1 for identity metrics and 5 for bit score.
id.splines	Smoothing parameter for the splines in the identity histogram. Zero (0) for no splines. A generally good value is 1/2. If non-zero, requires the stats package.
id.metric	Metric of identity to be used (Y-axis). It can be any unambiguous prefix of "identity", "corrected identity", or "bit score".
id.summary	Method used to build the identity histogram (Horizontal axis of the right panel). It can be any unambiguous prefix of "sum", "average", "median", "90% lower bound", "90% upper bound", "95% lower bound", and "95% upper bound". The last four options correspond to the upper and lower boundaries of the 90% and 95% empirical confidence intervals.
pos.min	Minimum (leftmost) position in the reference (concatenated) genome (in bp).
pos.max	Maximum (rightmost) position in the reference (concatenated) genome (in bp). By default: Length of the genome.
pos.binsize	Size of the position bins (horizontal histograms) in bp.
pos.splines	Smoothing parameter for the splines in the position histogram. Zero (0) for no splines. If non-zero, requires the stats package.
rec.col1	Lightest color in the recruitment plot.
rec.col2	Darkest color in the recruitment plot.

<code>main</code>	Title of the plot.
<code>contig.col</code>	Color of the Contig boundaries. Set to NA to ignore Contig boundaries.
<code>ret.recplot</code>	Indicates if the matrix of the recruitment plot is to be returned.
<code>ret.hist</code>	Ignored, for backwards compatibility.
<code>ret.mode</code>	Indicates if the mode of the identity is to be computed. It requires the <code>modeest</code> package.
<code>id.cutoff</code>	Minimum identity to consider an alignment as "top". By default, it is 0.95 for the identity metrics and 95% of the best scoring alignment for bit score.
<code>verbose</code>	Indicates if the function should report the advance.
<code>...</code>	Any additional graphic parameters to be passed to <code>plot</code> for all panels except the recruitment plot (lower-left).

Value

A list with the following elements:

`pos.marks`: Midpoints of the position histogram.

`id.matrix`: Midpoints of the identity histogram.

`recplot` (if `ret.recplot=TRUE`): Matrix containing the recruitment plot values.

`id.mean`: Mean identity.

`id.median`: Median identity.

`id.mode` (if `ret.mode=TRUE`): Mode of the identity. Deprecated.

`id.hist` (if `ret.hist=TRUE`): Values of the identity histogram.

`pos.hist.low` (if `ret.hist=TRUE`): Values of the position histogram (depth) with "low" identity (i.e., below `id.cutoff`).

`pos.hist.top` (if `ret.hist=TRUE`): Values of the position histogram (depth) with "top" identity (i.e., above `id.cutoff`).

`id.max`: Value of `id.max`. This is returned because `id.max=NULL` may vary.

`id.cutoff`: Value of `id.cutoff`. This is returned because `id.cutoff=NULL` may vary.

`seqdepth.mean.top`: Average sequencing depth with identity above `id.cutoff`.

`seqdepth.mean.low`: Average sequencing depth with identity below `id.cutoff`.

`seqdepth.mean.all`: Average sequencing depth without identity filtering.

`seqdepth.median.top`: Median sequencing depth with identity above `id.cutoff`.

`seqdepth.median.low`: Median sequencing depth with identity below `id.cutoff`.

`seqdepth.median.all`: Median sequencing depth without identity filtering.

`id.metric`: Full name of the used identity metric.

`id.summary`: Full name of the summary method used to build the identity plot.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2 *enve recplot2*

Description

Produces recruitment plots provided that BlastTab.catsbj.pl has been previously executed.

Usage

```
enve.recplot2(prefix, plot = TRUE, pos.breaks = 1000, pos.breaks.tsv = NA,
  id.breaks = 300, id.free.range = FALSE, id.metric = c("identity",
    "corrected identity", "bit score"), id.summary = sum,
  id.cutoff = 95, threads = 2, verbose = TRUE, ...)
```

Arguments

prefix	Path to the prefix of the BlastTab.catsbj.pl output files. At least the files .rec and .lim must exist with this prefix.
plot	Should the object be plotted?
pos.breaks	Breaks in the positions histogram. It can also be a vector of break points, and values outside the range are ignored. If zero (0), it uses the sequence breaks as defined in the .lim file, which means one bin per contig (or gene, if the mapping is against genes). Ignored if 'pos.breaks.tsv' is passed.
pos.breaks.tsv	Path to a list of (absolute) coordinates to use as position breaks. This tab-delimited file can be produced by 'GFF.catsbj.pl', and it must contain at least one column: coordinates of the break positions of each position bin. If it has a second column, this is used as the name of the position bin that ends at the given coordinate (the first row is ignored). Any additional columns are currently ignored. If NA, position bins are determined by 'pos.breaks'.
id.breaks	Breaks in the identity histogram. It can also be a vector of break points, and values outside the range are ignored.
id.free.range	Indicates that the range should be freely set from the observed values. Otherwise, 70-100% is included in the identity histogram (default).
id.metric	Metric of identity to be used (Y-axis). Corrected identity is only supported if the original BLAST file included sequence lengths.
id.summary	Function summarizing the identity bins. Other recommended options include: 'median' to estimate the median instead of total bins, and 'function(x) mlv(x,method='parzen')\$M' to estimate the mode.
id.cutoff	Cutoff of identity metric above which the hits are considered 'in-group'. The 95% identity corresponds to the expectation of ANI<95% within species.
threads	Number of threads to use.
verbose	Indicates if the function should report the advance.
...	Any additional parameters supported by 'plot.enve.RecPlot2'.

Value

Returns an object of class 'enve.RecPlot2'.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.RecPlot2-class *enve.RecPlot2 S4 class*

Description

Enve-omics representation of Recruitment plots. This object can be produced by 'enve.recplot2' and supports S4 method plot.

Objects from the Class

Objects can be created by calls of the form `new(enve.RecPlot2 ...)`

Slots

`counts:` (matrix) Counts as a two-dimensional histogram.
`pos.counts.in:` (numeric) Counts of in-group hits per position bin.
`pos.counts.out:` (numeric) Counts of out-group hits per position bin.
`id.counts:` (numeric) Counts per ID bin.
`id.breaks:` (numeric) Breaks of identity bins.
`pos.breaks:` (numeric) Breaks of position bins.
`pos.names:` (character) Names of the position bins.
`seq.breaks:` (numeric) Breaks of input sequences.
`peaks:` (list) Peaks identified in the recplot. Limits of the subject sequences after concatenation.
`seq.names:` (character) Names of the subject sequences.
`id.metric:` (character) Metric used as 'identity'.
`id.ingroup:` (logical) Identity bins considered in-group.
`call:` (call) Call producing this object.

Methods

`$ signature(x = "enve.RecPlot2"): ...`

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.ANIR *enve recplot2 ANIR*

Description

Estimate the Average Nucleotide Identity from reads (ANIR) from a recruitment plot

Usage

```
enve.recplot2.ANIR(x, range = c(0, Inf))
```

Arguments

x	'enve.RecPlot2' object.
range	Range of identities to be considered. By default, the full range is used (note that the upper boundary is 'Inf' and not 100 because recruitment plots can also be built with bit-scores). To use only intra-population matches (with identities), use c(95,100). To use only inter-population values, use c(0,95).

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.changeCutoff
enve recplot2 changeCutoff

Description

Change the intra-species cutoff of an existing recruitment plot.

Usage

```
enve.recplot2.changeCutoff(rp, new.cutoff = 98)
```

Arguments

rp	enve.RecPlot2 object.
new.cutoff	New cutoff to use.

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.compareIdentities
      enve recplot2 compareIdentities
```

Description

Compare the distribution of identities between two enve.RecPlot2 objects.

Usage

```
enve.recplot2.compareIdentities(x, y, method = "hellinger", smooth.par = NULL,
  pseudocounts = 0, max.deviation = 0.75)
```

Arguments

x	First enve.RecPlot2 object.
y	Second enve.RecPlot2 object.
method	Distance method to use. This should be (an unambiguous abbreviation of) one of: "hellinger" (Hellinger, 1090, doi:10.1515/crll.1909.136.210), "bhattacharyya" (Bhattacharyya, 1943, Bull. Calcutta Math. Soc. 35), "kl" or "kullback-leibler" (Kullback & Leibler, 1951, doi:10.1214/aoms/1177729694), or "euclidean".
smooth.par	Smoothing parameter for cubic spline smoothing. Use 0 for no smoothing. Use NULL to automatically determine this value using leave-one-out cross-validation (see 'smooth.spline' parameter 'spar').
pseudocounts	Smoothing parameter for Laplace smoothing. Use 0 for no smoothing, or 1 for add-one smoothing.
max.deviation	Maximum mean deviation between identity breaks tolerated (as percent identity). Difference in number of id.breaks is never tolerated.

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.coordinates
      enve recplot2 coordinates
```

Description

Returns the sequence name and coordinates of the requested position bins.

Usage

```
enve.recplot2.coordinates(x, bins)
```

Arguments

x	'enve.RecPlot2' object.
bins	Vector of selected bins to return. It can be a vector of logical values with the same length as 'x\$pos.breaks'-1 or a vector of integers. If missing, returns the coordinates of all windows.

Value

Returns a data.frame with four columns: name.from (character), pos.from (numeric) name.to (character), and pos.to (numeric). The first two correspond to sequence and position of the start point of the bin, the last two correspond to the sequence and position of the end point of the bin.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.corePeak

enve recplot2 corePeak

Description

Finds the peak in a list of peaks that is most likely to represent the "core genome" of a population.

Usage

enve.recplot2.corePeak(x)

Arguments

x	'list' of 'enve.RecPlot2.Peak' objects.
---	---

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.extractWindows  
  enve recplot2 extractWindows
```

Description

Extract windows significantly below (or above) the peak in sequencing depth.

Usage

```
enve.recplot2.extractWindows(rp, peak, lower.tail = TRUE, significance = 0.05,  
  seq.names = FALSE)
```

Arguments

<code>rp</code>	Recruitment plot, a 'enve.RecPlot2' object.
<code>peak</code>	Peak, an 'enve.RecPlot2.Peak' object. If list, it is assumed to be a list of 'enve.RecPlot2.Peak' objects, in which case the core peak is used (see 'enve.recplot2.corePeak').
<code>lower.tail</code>	If FALSE, it returns windows significantly above the peak in sequencing depth.
<code>significance</code>	Significance threshold (alpha) to select windows.
<code>seq.names</code>	Returns subject sequence names instead of a vector of Booleans. If the recruitment plot was generated with named position bins (e.g. using 'pos.breaks'=0 or a two-column 'pos.breaks.tsv'), it returns a vector of characters (the sequence identifiers), otherwise it returns a data.frame with a name column and two columns of coordinates.

Value

Returns a vector of logicals if 'seq.names=FALSE'. If 'seq.names=TRUE', it returns a vector of characters if the object has 'pos.names' defined, or a data.frame with four columns otherwise: name.from, name.to, pos.from, and pos.to (see 'enve.recplot2.coordinates').

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks
enve recplot2 findPeaks

Description

Identifies peaks in the population histogram potentially indicating sub-population mixtures

Usage

```
enve.recplot2.findPeaks(x, method = "emauto", ...)
```

Arguments

x	An 'enve.RecPlot2' object.
method	Peak-finder method. This should be one of: "emauto" (Expectation-Maximization with auto-selection of components), "em" (Expectation-Maximization), "mower" (Custom distribution-mowing method).
...	Any additional parameters supported by 'enve.recplot2.findPeaks.<method>'

Value

Returns a list of 'enve.RecPlot2.Peak' objects.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks.em
enve recplot2 findPeaks em

Description

Identifies peaks in the population histogram using a Gaussian Mixture Model Expectation Maximization (GMM-EM) method.

Usage

```
enve.recplot2.findPeaks.em(x, max.iter = 1000, ll.diff.res = 1e-08,  
  components = 2, rm.top = 0.05, verbose = FALSE, init, log = TRUE)
```

Arguments

<code>x</code>	An 'enve.RecPlot2' object.
<code>max.iter</code>	Maximum number of EM iterations.
<code>ll.diff.res</code>	Maximum Log-Likelihood difference to be considered as convergent.
<code>components</code>	Number of distributions assumed in the mixture.
<code>rm.top</code>	Top-values to remove before finding peaks, as a quantile probability. This step is useful to remove highly conserved regions, but can be turned off by setting <code>rm.top=0</code> . The quantile is determined <i>after</i> removing zero-coverage windows.
<code>verbose</code>	Display (mostly debugging) information.
<code>init</code>	Initialization parameters. By default, these are derived from k-means clustering. A named list with vectors for 'mu', 'sd', and 'alpha', each of length 'components'.
<code>log</code>	Logical value indicating if the estimations should be performed in natural logarithm units. Do not change unless you know what you're doing.

Value

Returns a list of 'enve.RecPlot2.Peak' objects.

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.findPeaks.emauto
      enve recplot2 findPeaks emauto
```

Description

Identifies peaks in the population histogram using a Gaussian Mixture Model Expectation Maximization (GMM-EM) method with number of components automatically detected.

Usage

```
enve.recplot2.findPeaks.emauto(x, components = seq(1, 10), criterion = "aic",
  merge.tol = 2L, verbose = FALSE, ...)
```

Arguments

<code>x</code>	An 'enve.RecPlot2' object.
<code>components</code>	A vector of number of components to evaluate.
<code>criterion</code>	Criterion to use for components selection. Must be one of: 'aic' (Akaike Information Criterion), 'bic' or 'sbc' (Bayesian Information Criterion or Schwarz Criterion).

merge.to1	When attempting to merge peaks with very similar sequencing depth, use this number of significant digits (in log-scale).
verbose	Display (mostly debugging) information.
...	Any additional parameters supported by 'enve.recplot2.findPeaks.em'.

Value

Returns a list of 'enve.RecPlot2.Peak' objects.

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.findPeaks.mower
      enve recplot2 findPeaks mower
```

Description

Identifies peaks in the population histogram potentially indicating sub-population mixtures, using a custom distribution-mowing method.

Usage

```
enve.recplot2.findPeaks.mower(x, min.points = 10, quant.est = c(0.002,
  0.998), mlv.opts = list(method = "parzen"), fitdist.opts.sn = list(distr = "sn",
  method = "qme", probs = c(0.1, 0.5, 0.8), start = list(omega = 1,
  alpha = -1), lower = c(0, -Inf, -Inf)), fitdist.opts.norm = list(distr = "norm",
  method = "qme", probs = c(0.4, 0.6), start = list(sd = 1),
  lower = c(0, -Inf)), rm.top = 0.05, with.skewness = TRUE,
  optim.rounds = 200, optim.epsilon = 1e-04, merge.logdist = log(1.75),
  verbose = FALSE, log = TRUE)
```

Arguments

x	An 'enve.RecPlot2' object.
min.points	Minimum number of points in the quantile-estimation-range ('quant.est') to estimate a peak.
quant.est	Range of quantiles to be used in the estimation of a peak's parameters.
mlv.opts	Ignored. For backwards compatibility.
fitdist.opts.sn	
fitdist.opts.norm	

rm.top	Top-values to remove before finding peaks, as a quantile probability. This step is useful to remove highly conserved regions, but can be turned off by setting rm.top=0. The quantile is determined <i>after</i> removing zero-coverage windows.
with.skewness	Allow skewness correction of the peaks. Typically, the sequencing-depth distribution for a single peak is left-skewed, due partly (but not exclusively) to fragmentation and mapping sensitivity. See Lindner et al 2013, <i>Bioinformatics</i> 29(10):1260-7 for an alternative solution for the first problem (fragmentation) called "tail distribution".
optim.rounds	Maximum rounds of peak optimization.
optim.epsilon	Trace change at which optimization stops (unless 'optim.rounds' is reached first). The trace change is estimated as the sum of square differences between parameters in one round and those from two rounds earlier (to avoid infinite loops from approximation).
merge.logdist	Maximum value of llog-ratio between centrality parameters in peaks to attempt merging. The default of ~0.22 corresponds to a maximum difference of 25%.
verbose	Display (mostly debugging) information.
log	Logical value indicating if the estimations should be performed in natural logarithm units. Do not change unless you know what you're doing.

Value

Returns a list of 'enve.RecPlot2.Peak' objects.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks.__emauto_one

enve recplot2 findPeaks emauto one

Description

Internal ancilliary function (see 'enve.recplot2.findPeaks.emauto').

Usage

```
enve.recplot2.findPeaks.__emauto_one(x, comp, do_crit, best,
  verbose, ...)
```

Arguments

x
comp
do_crit
best
verbose
...

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks.__em_e
enve recplot2 findPeaks em e

Description

Internal ancilliary function (see 'enve.recplot2.findPeaks.em').

Usage

```
enve.recplot2.findPeaks.__em_e(x, theta)
```

Arguments

x
theta

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks.__em_m
enve recplot2 findPeaks em m

Description

Internal ancilliary function (see ‘enve.recplot2.findPeaks.em’)

Usage

enve.recplot2.findPeaks.__em_m(x, posterior)

Arguments

x
posterior

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.findPeaks.__mower
enve recplot2 findPeaks mower

Description

Internal ancilliary function (see ‘enve.recplot2.findPeaks.mower’).

Usage

enve.recplot2.findPeaks.__mower(peaks.opts)

Arguments

peaks.opts

Author(s)

Luis M. Rodriguez-R [aut, cre]

```
enve.recplot2.findPeaks.__mow_one
      enve recplot2 findPeaks mow one
```

Description

Internall ancilliary function (see ‘enve.recplot2.findPeaks.mower’).

Usage

```
enve.recplot2.findPeaks.__mow_one(lsd1, min.points, quant.est,
  mlv.opts, fitdist.opts, with.skewness, optim.rounds, optim.epsilon,
  n.total, merge.logdist, verbose, log)
```

Arguments

lsd1
min.points
quant.est
mlv.opts
fitdist.opts
with.skewness
optim.rounds
optim.epsilon
n.total
merge.logdist
verbose
log

Author(s)

Luis M. Rodriguez-R [aut, cre]

 enve.RecPlot2.Peak-class

enve.RecPlot2.Peak S4 class

Description

Enve-omics representation of a peak in the sequencing depth histogram of a Recruitment plot (see ‘enve.recplot2.findPeaks’).

Objects from the Class

Objects can be created by calls of the form `new(enve.RecPlot2.Peak ...)`

Slots

dist: (character) Distribution of the peak. Currently supported: ‘norm’ (normal) and ‘sn’ (skew-normal).

values: (numeric) Sequencing depth values predicted to conform the peak.

values.res: (numeric) Sequencing depth values not explained by this or previously identified peaks.

mode: (numeric) Seed-value of mode anchoring the peak.

param.hat: (list) Parameters of the distribution. A list of two values if `dist=‘norm’` (sd and mean), or three values if `dist=‘sn’` (`omega=scale`, `alpha=shape`, and `xi=location`). Note that the "dispersion" parameter is always first and the "location" parameter is always last.

n.hat: (numeric) Number of bins estimated to be explained by this peak. This should ideally be equal to the length of ‘values’, but it’s not and integer.

n.total: (numeric) Total number of bins from which the peak was extracted. I.e., total number of position bins with non-zero sequencing depth in the recruitment plot (regardless of peak count).

err.res: (numeric) Error left after adding the peak (mower) or log-likelihood (em or emauto).

merge.logdist: (numeric) Attempted ‘merge.logdist’ parameter.

seq.depth: (numeric) Best estimate available for the sequencing depth of the peak (centrality).

log: (logical) Indicates if the estimation was performed in natural logarithm space

Methods

`$ signature(x = "enve.RecPlot2.Peak"): ...`

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.seqdepth

enve recplot2 seqdepth

Description

Calculate the sequencing depth of the given window(s)

Usage

```
enve.recplot2.seqdepth(x, sel, low.identity = FALSE)
```

Arguments

x	'enve.RecPlot2' object.
sel	Window(s) for which the sequencing depth is to be calculated. If not passed, it returns the sequencing depth of all windows
low.identity	A logical indicating if the sequencing depth is to be estimated only with low-identity matches. By default, only high-identity matches are used.

Value

Returns a numeric vector of sequencing depths (in bp/bp).

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.windowDepthThreshold

enve recplot2 windowDepthThreshold

Description

Identifies the threshold below which windows should be identified as variable or absent.

Usage

```
enve.recplot2.windowDepthThreshold(rp, peak, lower.tail = TRUE,  
  significance = 0.05)
```

Arguments

rp	Recruitment plot, an 'enve.RecPlot2' object.
peak	Peak, an 'enve.RecPlot2.Peak' object. If list, it is assumed to be a list of 'enve.RecPlot2.Peak' objects, in which case the core peak is used (see 'enve.recplot2.corePeak').
lower.tail	If FALSE, it returns windows significantly above the peak in sequencing depth.
significance	Significance threshold (alpha) to select windows.

Value

Returns a float. The units are depth if the peaks were estimated in linear scale, or log-depth otherwise ('peak\$log').

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.__counts

enve recplot2 counts

Description

Internal ancilliary function (see 'enve.recplot2').

Usage

```
enve.recplot2.__counts(x, pos.breaks, id.breaks, rec.idcol)
```

Arguments

x
pos.breaks
id.breaks
rec.idcol

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.__peakHist
enve recplot2 peakHist

Description

Internal ancilliary function (see 'enve.RecPlot2.Peak').

Usage

```
enve.recplot2.__peakHist(x, mids, counts = TRUE)
```

Arguments

x
mids
counts

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.recplot2.__whichClosestPeak
enve recplot2 whichClosestPeak

Description

Internal ancilliary function (see 'enve.recplot2.findPeaks').

Usage

```
enve.recplot2.__whichClosestPeak(peak, peaks)
```

Arguments

peak
peaks

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.tribs

enve tribs

Description

Subsample any objects in "distance space" to reduce the effect of sample-clustering. This function was originally designed to subsample genomes in "phylogenetic distance space", a clear case of strong clustering bias in sampling, by Luis M. Rodriguez-R and Michael R Weigand.

Usage

```
enve.tribs(dist, selection = labels(dist), replicates = 1000,
  summary.fx = median, dist.method = "euclidean", subsamples = seq(0,
    1, by = 0.01), dimensions = ceiling(length(selection) *
    0.05), metaMDS.opts = list(), threads = 2, verbosity = 1,
  points, pre.tribs)
```

Arguments

dist	Distances as a 'dist' object.
selection	Objects to include in the subsample. By default, all objects are selected.
replicates	Number of replications per point
summary.fx	Function to summarize the distance distributions in a given replicate. By default, the median distance is estimated.
dist.method	Distance method between random points and samples in the transformed space. See 'dist'.
subsamples	Subsampling fractions
dimensions	Dimensions to use in the NMDS. By default, 5% of the selection length.
metaMDS.opts	Any additional options to pass to metaMDS, as 'list'.
threads	Number of threads to use.
verbosity	Verbosity. Use 0 to run quietly, increase for additional information.
points	Optional. If passed, the MDS step is skipped and this object is used instead. It can be the '\$points' slot of class 'metaMDS' (from 'vegan'). It must be a matrix or matrix-coercible object, with samples as rows and dimensions as columns.
pre.tribs	Optional. If passed, the points are recovered from this object (except if 'points' is also passed. This should be an 'enve.TRIBS' object estimated on the same objects (the selection is unimportant).

Value

Returns an 'enve.TRIBS' object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

`enve.TRIBS-class``enve.TRIBS S4 class`

Description

Enve-omics representation of "Transformed-space Resampling In Biased Sets (TRIBS)". This object represents sets of distances between objects, sampled nearly-uniformly at random in "distance space". Subsampling without selection is trivial, since both the distances space and the selection occur in the same transformed space. However, it's useful to compare randomly subsampled sets against a selected set of objects. This is intended to identify overdispersion or overclustering (see `'enve.TRIBStest'`) of a subset against the entire collection of objects with minimum impact of sampling biases. This object can be produced by `'enve.tribs'` and supports S4 methods `'plot'` and `'summary'`.

Objects from the Class

Objects can be created by calls of the form `new(enve.TRIBS . . .)`

Slots

`distance:` (numeric) Centrality measurement of the distances between the `«` selected objects (without subsampling).

`points:` (matrix) Position of the different objects in distance `«` space.

`distances:` (matrix) Subsampled distances, where the rows are replicates `«` and the columns are subsampling levels.

`spaceSize:` (numeric) Number of objects.

`selSize:` (numeric) Number of selected objects.

`dimensions:` (numeric) Number of dimensions in the distance space.

`subsamples:` (numeric) Subsampling levels (as fractions, from 0 to 1).

`call:` (call) Call producing this object.

Methods

No methods defined with class "enve.TRIBS" in the signature.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.TRIBS.merge *enve TRIBS merge*

Description

Merges two ‘enve.TRIBS’ objects generated from the same objects at different subsampling levels.

Usage

```
enve.TRIBS.merge(x, y)
```

Arguments

x	First ‘enve.TRIBS’ object.
y	Second ‘enve.TRIBS’ object.

Value

Returns an ‘enve.TRIBS’ object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.tribs.test *enve tribs test*

Description

Estimates the empirical difference between all the distances in a set of objects and a subset, together with its statistical significance.

Usage

```
enve.tribs.test(dist, selection, bins = 50, ...)
```

Arguments

dist	Distances as ‘dist’ object.
selection	Selection defining the subset.
bins	Number of bins to evaluate in the range of distances.
...	Any other parameters supported by ‘enve.tribs’, except ‘subsamples’.

Value

Returns an ‘enve.TRIBStest’ object.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.TRIBStest-class *enve.TRIBStest S4 class*

Description

Test of significance of overclustering or overdispersion in a selected set of objects with respect to the entire set (see ‘enve.TRIBS’). This object can be produced by ‘enve.tribs.test’ and supports S4 methods ‘plot’ and ‘summary’.

Objects from the Class

Objects can be created by calls of the form `new(enve.TRIBStest ...)`

Slots

`pval.gt`: (numeric) P-value for the overdispersion test.

`pval.lt`: (numeric) P-value for the overclustering test.

`all.dist`: (numeric) Empiric PDF of distances for the entire dataset (subsampled at selection size).

`sel.dist`: (numeric) Empiric PDF of distances for the selected objects (without subsampling).

`diff.dist`: (numeric) Empiric PDF of the difference between ‘all.dist’ and ‘sel.dist’. The p-values are estimating by comparing areas in this PDF greater than and lesser than zero.

`dist.mids`: (numeric) Midpoints of the empiric PDFs of distances.

`diff.mids`: (numeric) Midpoints of the empiric PDF of difference of distances.

`call`: (call) Call producing this object.

Methods

No methods defined with class "enve.TRIBStest" in the signature.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.truncate	<i>enve truncate</i>
---------------	----------------------

Description

Removes the 'n' highest and lowest values from a vector, and applies a summary function. The value of 'n' is determined such that the central range is used, corresponding to the 'f' fraction of values.

Usage

```
enve.truncate(x, f = 0.95, FUN = mean)
```

Arguments

x	A vector of numbers.
f	The fraction of values to retain.
FUN	Summary function to apply to the vectors. To obtain the truncated vector itself, use 'c'.

Value

Returns the summary ('FUN') of the truncated vector.

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.__prune.iter	<i>enve prune iter</i>
-------------------	------------------------

Description

Internal function for enve.prune.dist

Usage

```
enve.__prune.iter(t, dist, min_dist, quiet)
```

Arguments

t
dist
min_dist
quiet

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.__prune.reduce *enve prune reduce*

Description

Internal function for enve.prune.dist

Usage

```
enve.__prune.reduce(t, nodes, min_dist, quiet)
```

Arguments

t
nodes
min_dist
quiet

Author(s)

Luis M. Rodriguez-R [aut, cre]

enve.__tribs *enve tribs*

Description

Internal ancilliary function (see 'enve.tribs').

Usage

```
enve.__tribs(rep, frx, selection, dimensions, dots, dist.method,  
             summary.fx, dist)
```

Arguments

rep
frx
selection
dimensions
dots
dist.method
summary.fx
dist

Author(s)

Luis M. Rodriguez-R [aut, cre]

growth.curves

Bacterial growth curves for three Escherichia coli mutants

Description

This data set provides time (first column) and three triplicated growth curves as optical density at 600nm (OD_600nm) for different mutants of E. coli.

Usage

growth.curves

Format

A data frame with 16 rows (times) and 10 rows (times and OD_600nm).

phyla.counts

Counts of microbial phyla in four sites

Description

This data set gives the counts of phyla in three different sites.

Usage

phyla.counts

Format

A data frame with 9 rows (phyla) and 4 rows (sites).

plot.enve.GrowthCurve *plot enve GrowthCurve*

Description

Plots an 'enve.GrowthCurve' object.

Usage

```
## S3 method for class 'enve.GrowthCurve'
plot(x, col, pt.alpha = 0.9, ln.alpha = 1, ln.lwd = 1,
     ln.lty = 1, band.alpha = 0.4, band.density = NULL, band.angle = 45,
     xp.alpha = 0.5, xp.lwd = 1, xp.lty = 1, pch = 19, new = TRUE,
     legend = new, add.params = FALSE, ...)
```

Arguments

x	'enve.GrowthCurve' object to plot.
col	Base colors to use for the different samples. Can be recycled. By default, grey for one sample or rainbow colors for more than one.
pt.alpha	Color alpha for the observed data points, using 'col' as a base.
ln.alpha	Color alpha for the fitted growth curve, using 'col' as a base.
ln.lwd	Line width for the fitted curve.
ln.lty	Line type for the fitted curve.
band.alpha	Color alpha for the confidence interval band of the fitted growth curve, using 'col' as a base.
band.density	Density of the filling pattern in the interval band. If NULL, a solid color is used.
band.angle	Angle of the density filling pattern in the interval band. Ignored if 'band.density' is NULL.
xp.alpha	Color alpha for the line connecting individual experiments, using 'col' as a base.
xp.lwd	Width of line for the experiments.
xp.lty	Type of line for the experiments.
pch	Point character for observed data points.
new	Should a new plot be generated? If FALSE, the existing canvas is used.
legend	Should the plot include a legend? If FALSE, no legend is added. If TRUE, a legend is added in the bottom-right corner. Otherwise, a legend is added in the position specified as 'xy.coords'.
add.params	Should the legend include the parameters of the fitted model?
...	Any other graphic parameters.

Author(s)

Luis M. Rodriguez-R [aut, cre]

plot.enve.RecPlot2 *plot enve RecPlot2*

Description

Plots an ‘enve.RecPlot2’ object.

Usage

```
## S3 method for class 'enve.RecPlot2'
plot(x, layout = matrix(c(5, 5, 2, 1, 4, 3), nrow = 2),
     panel.fun = list(), widths = c(1, 7, 2), heights = c(1, 2),
     palette = grey((100:0)/100), underlay.group = TRUE, peaks.col = "darkred",
     use.peaks, id.lim = range(x$id.breaks), pos.lim = range(x$pos.breaks),
     pos.units = c("Mbp", "Kbp", "bp"), mar = list(`1` = c(5,
     4, 1, 1) + 0.1, `2` = c(ifelse(any(layout == 1), 1, 5),
     4, 4, 1) + 0.1, `3` = c(5, ifelse(any(layout == 1), 1,
     4), 1, 2) + 0.1, `4` = c(ifelse(any(layout == 1), 1,
     5), ifelse(any(layout == 2), 1, 4), 4, 2) + 0.1, `5` = c(5,
     3, 4, 1) + 0.1, `6` = c(5, 4, 4, 2) + 0.1), pos.splines = 0,
     id.splines = 1/2, in.lwd = ifelse(is.null(pos.splines) ||
     pos.splines > 0, 1/2, 2), out.lwd = ifelse(is.null(pos.splines) ||
     pos.splines > 0, 1/2, 2), id.lwd = ifelse(is.null(id.splines) ||
     id.splines > 0, 1/2, 2), in.col = "darkblue", out.col = "lightblue",
     id.col = "black", breaks.col = "#AAAAAA40", peaks.opts = list(),
     ...)
```

Arguments

x	‘enve.RecPlot2’ object to plot.
layout	Matrix indicating the position of the different panels in the layout, where: 0: Empty space, 1: Counts matrix, 2: position histogram (sequencing depth), 3: identity histogram, 4: Populations histogram (histogram of sequencing depths), 5: Color scale for the counts matrix (vertical), 6: Color scale of the counts matrix (horizontal) Only panels indicated here will be plotted. To plot only one panel simply set this to the number of the panel you want to plot.
panel.fun	List of functions to be executed after drawing each panel. Use the indices in ‘layout’ (as characters) as keys. Functions for indices missing in ‘layout’ are ignored. For example, to add a vertical line at the 3Mbp mark in both the position histogram and the counts matrix: ‘list(`1`=function() abline(v=3), `2`=function() abline(v=3))’. Note that the X-axis in both panels is in Mbp by default. To change this behavior, set ‘pos.units’ accordingly.
widths	Relative widths of the columns of ‘layout’.
heights	Relative heights of the rows of ‘layout’.
palette	Colors to be used to represent the counts matrix, sorted from no hits to the maximum sequencing depth.

<code>underlay.group</code>	If TRUE, it indicates the in-group and out-group areas coloured based on ‘in.col’ and ‘out.col’. Requires support for semi-transparency.
<code>peaks.col</code>	If not NA, it attempts to represent peaks in the population histogram in the specified color. Set to NA to avoid peak-finding.
<code>use.peaks</code>	A list of ‘enve.RecPlot2.Peak’ objects, as returned by ‘enve.recplot2.findPeaks’. If passed, ‘peaks.opts’ is ignored.
<code>id.lim</code>	Limits of identities to represent.
<code>pos.lim</code>	Limits of positions to represent (in bp, regardless of ‘pos.units’).
<code>pos.units</code>	Units in which the positions should be represented (powers of 1,000 base pairs).
<code>mar</code>	
<code>pos.splines</code>	Smoothing parameter for the splines in the position histogram. Zero (0) for no splines. Use NULL to automatically detect by leave-one-out cross-validation.
<code>id.splines</code>	Smoothing parameter for the splines in the identity histogram. Zero (0) for no splines. Use NULL to automatically detect by leave-one-out cross-validation.
<code>in.lwd</code>	Line width for the sequencing depth of in-group matches.
<code>out.lwd</code>	Line width for the sequencing depth of out-group matches.
<code>id.lwd</code>	Line width for the identity histogram.
<code>in.col</code>	Color associated to in-group matches.
<code>out.col</code>	Color associated to out-group matches.
<code>id.col</code>	Color for the identity histogram.
<code>breaks.col</code>	Color of the vertical lines indicating sequence breaks.
<code>peaks.opts</code>	Options passed to ‘enve.recplot2.findPeaks’, if ‘peaks.col’ is not NA.
<code>...</code>	Any other graphic parameters (currently ignored).

Value

Returns a list of ‘enve.RecPlot2.Peak’ objects (see ‘enve.recplot2.findPeaks’). If ‘peaks.col’=NA or ‘layout’ doesn’t include 4, returns NA.

Author(s)

Luis M. Rodriguez-R [aut, cre]

`plot.enve.TRIBS`

plot enve TRIBS

Description

Plot an ‘enve.TRIBS’ object.

Usage

```
## S3 method for class 'enve.TRIBS'
plot(x, new = TRUE, type = c("boxplot", "points"),
     col = "#00000044", pt.cex = 1/2, pt.pch = 19, pt.col = col,
     ln.col = col, ...)
```

Arguments

x	'enve.TRIBS' object to plot.
new	Should a new canvas be drawn?
type	Type of plot. The 'points' plot shows all the replicates, the 'boxplot' plot represents the values found by 'boxplot.stats' as areas, and plots the outliers as points.
col	Color of the areas and/or the points.
pt.cex	Size of the points.
pt.pch	Points character.
pt.col	Color of the points.
ln.col	Color of the lines.
...	Any additional parameters supported by 'plot'.

Author(s)

Luis M. Rodriguez-R [aut, cre]

plot.enve.TRIBStest *plot enve TRIBStest*

Description

Plots an 'enve.TRIBStest' object.

Usage

```
## S3 method for class 'enve.TRIBStest'
plot(x, type = c("overlap", "difference"), col = "#00000044",
     col1 = col, col2 = "#44001144", ylab = "Probability", xlim = range(attr(x,
     "dist.mids")), ylim = c(0, max(c(attr(x, "all.dist"),
     attr(x, "sel.dist")))), ...)
```

Arguments

x	'enve.TRIBStest' object to plot.
type	What to plot. 'overlap' generates a plot of the two contrasting empirical PDFs (to compare against each other), 'difference' produces a plot of the differences between the empirical PDFs (to compare against zero).
col	Main color of the plot if type='difference'.
col1	First color of the plot if type='overlap'.
col2	Second color of the plot if type='overlap'.
ylab	Y-axis label.
xlim	X-axis limits.
ylim	Y-axis limits.
...	Any other graphical arguments.

Author(s)

Luis M. Rodriguez-R [aut, cre]

summary.enve.GrowthCurve
summary enve GrowthCurve

Description

Summary of an 'enve.GrowthCurve' object.

Usage

```
## S3 method for class 'enve.GrowthCurve'  
summary(object, ...)
```

Arguments

object	'enve.GrowthCurve' object.
...	No additional parameters are currently supported.

Author(s)

Luis M. Rodriguez-R [aut, cre]

summary.enve.TRIBS *summary enve TRIBS*

Description

Summary of an ‘enve.TRIBS’ object.

Usage

```
## S3 method for class 'enve.TRIBS'  
summary(object, ...)
```

Arguments

object ‘enve.TRIBS’ object.
... No additional parameters are currently supported.

Author(s)

Luis M. Rodriguez-R [aut, cre]

summary.enve.TRIBStest
 summary enve TRIBStest

Description

Summary of an ‘enve.TRIBStest’ object.

Usage

```
## S3 method for class 'enve.TRIBStest'  
summary(object, ...)
```

Arguments

object ‘enve.TRIBStest’ object.
... No additional parameters are currently supported.

Author(s)

Luis M. Rodriguez-R [aut, cre]

\$-methods

~~ Methods for Function \$ ~~

Description

~~ Methods for function \$ ~~

Methods

signature(x = "enve.GrowthCurve") enve.GrowthCurve, a growth curve

signature(x = "enve.RecPlot2") enve.RecPlot2, a recruitment plot

signature(x = "enve.RecPlot2.Peak") enve.RecPlot2.Peak, a peak from a recruitment plot

Index

- *Topic **datasets**
 - growth.curves, 37
 - phyla.counts, 37
- *Topic **methods**
 - \$-methods, 44
- *Topic **package**
 - enveomics.R-package, 3
- \$, enve.GrowthCurve-method (\$-methods), 44
- \$, enve.RecPlot2-method (\$-methods), 44
- \$, enve.RecPlot2.Peak-method (\$-methods), 44
- \$-methods, 44

- enve.__prune.iter, 35
- enve.__prune.reduce, 36
- enve.__tribs, 36
- enve.barplot, 3
- enve.cliopts, 4
- enve.col.alpha, 5
- enve.col2alpha, 6
- enve.df2dist, 6
- enve.df2dist.group, 7
- enve.df2dist.list, 8
- enve.GrowthCurve
 - (enve.GrowthCurve-class), 10
- enve.growthcurve, 9
- enve.GrowthCurve-class, 10
- enve.prune.dist, 11
- enve.recplot, 12
- enve.RecPlot2 (enve.RecPlot2-class), 15
- enve.recplot2, 14
- enve.RecPlot2-class, 15
- enve.recplot2.__counts, 29
- enve.recplot2.__peakHist, 30
- enve.recplot2.__whichClosestPeak, 30
- enve.recplot2.ANIr, 16
- enve.recplot2.changeCutoff, 16
- enve.recplot2.compareIdentities, 17
- enve.recplot2.coordinates, 17

- enve.recplot2.corePeak, 18
- enve.recplot2.extractWindows, 19
- enve.recplot2.findPeaks, 20
- enve.recplot2.findPeaks.__em_e, 24
- enve.recplot2.findPeaks.__em_m, 25
- enve.recplot2.findPeaks.__emauto_one, 23
- enve.recplot2.findPeaks.__mow_one, 26
- enve.recplot2.findPeaks.__mower, 25
- enve.recplot2.findPeaks.em, 20
- enve.recplot2.findPeaks.emauto, 21
- enve.recplot2.findPeaks.mower, 22
- enve.RecPlot2.Peak
 - (enve.RecPlot2.Peak-class), 27
- enve.RecPlot2.Peak-class, 27
- enve.recplot2.seqdepth, 28
- enve.recplot2.windowDepthThreshold, 28
- enve.TRIBS (enve.TRIBS-class), 32
- enve.tribs, 31
- enve.TRIBS-class, 32
- enve.TRIBS.merge, 33
- enve.tribs.test, 33
- enve.TRIBStest (enve.TRIBStest-class), 34
- enve.TRIBStest-class, 34
- enve.truncate, 35
- enveomics.R (enveomics.R-package), 3
- enveomics.R-package, 3

- growth.curves, 37

- phyla.counts, 37
- plot.enve.GrowthCurve, 38
- plot.enve.RecPlot2, 39
- plot.enve.TRIBS, 40
- plot.enve.TRIBStest, 41

- summary.enve.GrowthCurve, 42
- summary.enve.TRIBS, 43
- summary.enve.TRIBStest, 43