Package 'microSTASIS'

September 14, 2021

Title Microbiota STability ASsessment via Iterative cluStering

Version 0.1.0

Description The toolkit 'µSTASIS' has been developed for the stability analysis of microbiota in a temporal framework by leveraging on iterative clustering. Concretely, the core function uses Hartigan-Wong k-means algorithm as many times as possible for stressing out paired samples from the same individuals to test if they remain together for multiple numbers of clusters over a whole data set of individuals. Moreover, the package includes multiple functions to subset samples from paired times, validate the results or visualize the output.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports crayon, fmsb, future, future.apply, ggplot2, ggside, progressr, reshape2, stats, stringr

Depends R (>= 2.10)

NeedsCompilation no

Author Pedro Sánchez-Sánchez [aut, cre] (<https://orcid.org/0000-0002-4846-1813>), Alfonso Benítez-Páez [aut] (<https://orcid.org/0000-0001-5707-4340>)

Maintainer Pedro Sánchez-Sánchez <bio.pedro.technology@gmail.com>

Repository CRAN

Date/Publication 2021-09-14 06:40:09 UTC

R topics documented:

`clr` *Detected ASV from multiple individuals at four different sampling times.*

Description

A dataset containing the amplicon sequence variants of 131 samples from the gut microbiota of 43 individuals. The values are transformed from counts by applying centred log-transformation (CLR).

Usage

```
clr
```

Format

A data frame with 131 rows and 226 variables

`CV_iterative_clustering` *Cross validation of the iterative Hartigan-Wong k-means clustering.*

Description

Perform cross validation in the way of leave-one-out (LOO) or k-fold of the stability results from `iterative_clustering()`.

Usage

```
CV_iterative_clustering(data, results, common, k = 1L, parallel = TRUE)
```

Arguments

`data`	input matrix with paired times, i.e. samples to be stressed to multiple iterations.
`results`	the `stabilitas()` output for the concrete paired times used for validation.
`common`	pattern that separates the ID and the sampling time.
`k`	integer; number of individuals to subset from the data for each time running `iterative_clustering()`.
`parallel`	logical; FALSE to sequentially run the internal loop or TRUE to do it by parallel computing (number of cores = 4).

Value

Multiple lists with multiple objects of class "kmeans".

Examples

```
t1_t2 <- paired_times(data = clr, first = "_1",
                      second = "_25", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
cv_klist_t1_t2_k2 <- CV_iterative_clustering(data = t1_t2, results = result_t1_t2,
                                             common = "_0_", k = 2L, parallel = FALSE)
```

CV_results *Compute the error or plot the stability values after* CV_iterative_clustering().

Description

Compute the mean absolute error after the cross validation or plot lines connecting the stability values for each subset of the original matrix of paired times.

Usage

```
CV_results(
  data,
  cv_klist,
  output = "MAE",
  points = TRUE,
  k = 1L,
  size_line = 0.5
)
```

Arguments

data	input matrix with paired times, i.e. samples to be stressed to multiple iterations.
cv_klist	list resulting from CV_iterative_clustering().
output	character: MAE or viz; to return a data frame with the MAE or to visualize a line plot.
points	logical; if plotting, FALSE to only plot lines and TRUE to add points on the original stability value, i.e. result from stabilitas().
k	integer; number of individuals to subset from the data. The same as used in CV_iterative_clustering().
size_line	numeric; if plotting, size of the multiple lines.

Value

A vector with MAE values or a line plot in the form of a "ggplot" object with the values of stability for the multiple subsets and the original matrix of paired samples.

Examples

```
t1_t2 <- paired_times(data = clr, first = "_1",
                      second = "_25", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
cv_klist_t1_t2_k2 <- CV_iterative_clustering(data = t1_t2, results = result_t1_t2,
                                             common = "_0_", k = 2L, parallel = FALSE)
MAE_t1_t2 <- CV_results(data = t1_t2, cv_klist = cv_klist_t1_t2_k2,
                        output = "MAE", k = 2L)
MAE <- st_previz(results = list(MAE_t1_t2),
                 times = "MAE_t1_t2")
st_heatmap(data = MAE, label = TRUE,
           high = 'red2', midpoint = 0.075, low = 'forestgreen')
CV_results(data = t1_t2, cv_klist = cv_klist_t1_t2_k2,
           output = "viz", k = 2L)
```

---

iterative_clustering *Iterative Hartigan-Wong k-means clustering.*

Description

Perform Hartigan-Wong `stats::kmeans()` algorithm as many times as possible. The values of k are from 2 to the number of rows minus 1.

Usage

```
iterative_clustering(data, parallel = TRUE)
```

Arguments

`data`	input matrix with paired times, i.e. samples to be stressed to multiple iterations.
`parallel`	logical; FALSE to sequentially run the internal loop or TRUE to do it by parallel computing (number of cores = 4).

Value

A list with multiple objects of class "kmeans".

Examples

```
t1_t2 <- paired_times(data = clr[,1:50], first = "_1",
                      second = "_25", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
```

`metadata_groups` *Easily extract groups of individuals from metadata variables.*

Description

Easily extract groups of individuals from metadata variables.

Usage

```
metadata_groups(metadata, samples, individuals, col_number)
```

Arguments

`metadata`	data frame with one column of samples matching with the rownames of the original input matrix to `paired_times()`.
`samples`	vector from metadata corresponding to the samples ID.
`individuals`	vector of individuals located in the first column of the `st_previz()` output.
`col_number`	number of the column with the variable used for grouping individuals.

Value

A vector with the same length as the number of rows in the `st_previz()` output.

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
metadata <- data.frame(Sample = rownames(clr),
                       age = c(rep("youth", 65), rep("old", 131-65)))
group <- metadata_groups(metadata = metadata, samples = metadata$Sample,
                         individuals = results$individual, col_number = 2)
```

`paired_times` *Generate a subset matrix with paired times.*

Description

Generate a subset matrix with paired times.

Usage

```
paired_times(data, first, second, common)
```

Arguments

`data`	input matrix with each rowname including an ID, a common pattern and a sampling time.
`first`	pattern associated with the first of the two sampling times.
`second`	pattern associated with the second of the two sampling times.
`common`	pattern that separates the ID and the sampling time.

Value

A matrix with the same number of columns as input and the samples from both samples times.

Examples

```
t1_t2 <- paired_times(data = clr, first = "_1",
                      second = "_25", common = "_0_")
```

`pre_radarPC` *Compute the PCA to the* `paired_times()` *output previously to visualize radar charts of principal components.*

Description

Compute the PCA to the `paired_times()` output previously to visualize radar charts of principal components.

Usage

```
pre_radarPC(data, exp.var = 0.75, limit = 1.5)
```

Arguments

`data`	input matrix with paired times.
`exp.var`	desired explained variance, i.e. a value between 0 and 1.
`limit`	times to distance an interquartile range from the first and third quartile; default is 1.5, i.e. the whiskers of a boxplot.

Value

The needed objects to later plot a radar chart (or spider plot) of principal components.

Examples

```
t1_t2 <- paired_times(data = clr, first = "_1",
                      second = "_25", common = "_0_")
pre_radarPC(data = t1_t2, exp.var = 0.85)
```

`radarPC` *Plot a radar chart of paired samples from one individual or from different groups.*

Description

Plot a radar chart of paired samples from one individual or from different groups.

Usage

```
radarPC(
  data,
  samples = c(1, 2),
  pre_radar = pre_radar,
  legend = expression("t"[1], "t"[2]),
  colors = c("orange", "royalblue"),
  groups = NULL,
  fun = "mean"
)
```

Arguments

`data`	input matrix with paired times.
`samples`	position of the samples from an individual to be plotted.
`pre_radar`	list; the output from `pre_radarPC()`.
`legend`	text to plot.
`colors`	specify desired colors.
`groups`	vector with the same length as the number of rows in the `st_previz()` output.
`fun`	character: mean or median; if the lines should correspond to any of those statistics when adding a groups vector.

Value

A radar chart (or spider plot) of the principal components for two samples from the same individual or for two groups.

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
pre_radar <- pre_radarPC(data = t1_t2, exp.var = 0.85)
radarPC(data = t1_t2, samples = c(3,4), pre_radar = pre_radar, colors = c(1,2))
metadata <- data.frame(Sample = rownames(clr),
                       age = c(rep("youth", 65), rep("old", 131-65)))
group <- metadata_groups(metadata = metadata, samples = metadata$Sample,
                         individuals = results$individual, col_number = 2)
radarPC(data = t1_t2, pre_radar = pre_radar, groups = rep(group, each = 2),
        colors = c("cyan1", "cyan3", "brown1", "brown3"))
```

`stabilitas` *Extract the stability results of individuals from the output of* `iterative_clustering()`.

Description

Those individuals whose samples are clustered under the same label in list[[x]]$cluster sum 1. If samples are in different clusters sum 0. Then, this is done for all possible values of k and, finally, divided the sum by k, so obtaining a value between 0 and 1.

Usage

```
stabilitas(klist, common)
```

Arguments

`klist`	input list corresponding to the output of `iterative_clustering()`.
`common`	pattern that separates the ID and the sampling time.

Value

µSTASIS stability score (mS) for the individuals from the two selected sampling times.

Examples

```
t1_t2 <- paired_times(data = clr[,1:50], first = "_1",
                      second = "_25", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
```

st_evolution *Generate boxplots of the stability evolution throughout sampling times by groups*

Description

Generate boxplots of the stability evolution throughout sampling times by groups

Usage

```
st_evolution(data, groups = NULL, points = TRUE, linetype = 2)
```

Arguments

data	input data frame resulting from st_previz().
groups	vector with the same length as the number of rows in the st_previz() output.
points	logical; FALSE to only visualize boxplots or TRUE to also add individual points.
linetype	numeric; type of line to connect the median value of paired times; 0 to avoid the line.

Value

A plot with as many boxes as paired times by group in the form of a "ggplot" object.

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
metadata <- data.frame(Sample = rownames(clr),
                       age = c(rep("youth", 65), rep("old", 131-65)))
group <- metadata_groups(metadata = metadata, samples = metadata$Sample,
                         individuals = results$individual, col_number = 2)
st_evolution(results, groups = group, points = TRUE, linetype = 0)
```

st_heatmap *Plot a heatmap of the stability results*

Description

Plot a heatmap of the stability results

Usage

```
st_heatmap(
  data,
  order = "none",
  label = FALSE,
  low = "red2",
  mid = "yellow",
  high = "forestgreen",
  midpoint = 0.5
)
```

Arguments

data	input data frame resulting from st_previz().
order	character: none, mean or median; if the heatmap should be sorted by any of those statistics of the stability values by individuals.
label	logical; FALSE to avoid printing the value or TRUE to print it.
low	color for the lowest value.
mid	color for the middle value.
high	color for the highest values.
midpoint	value to situate the middle.

Value

A heatmap of the stability values in the form of a "ggplot" object

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
st_heatmap(data = results, order = "mean", label = TRUE)
```

st_previz *Process the stabilitas() output to a new format ready for the visualization functions.*

Description

Process the stabilitas() output to a new format ready for the visualization functions.

Usage

```
st_previz(results, times)
```

Arguments

results	a list with the stabilitas() outputs.
times	a vector with the names of each paired time, e.g. "t1_t2".

Value

A data frame ready for its use under the visualization functions.

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
```

st_scatter *Plot a scatter and side boxplot of the stability results*

Description

Plot a scatter and side boxplot of the stability results

Usage

```
st_scatter(data, order = "none", times, grid_lines = FALSE, side_scale = 0.3)
```

Arguments

data	input data frame resulting from st_previz().
order	character: none, mean or median; if the scatter plot should be sorted by any of those statistics of the stability values by individuals.
times	a vector with the names of each paired time, e.g. "t1_t2".
grid_lines	logical; FALSE to print a blank background or TRUE to include a gray grid
side_scale	numeric; scale of the side boxplot

Value

A scatter plot and a side boxplot of the stability values in the form of a "ggplot" object.

Examples

```
t1_t2 <- paired_times(data = clr[,1:25], first = "_1",
                      second = "_25", common = "_0_")
t2_t3 <- paired_times(data = clr[,1:25], first = "_25",
                      second = "_26", common = "_0_")
klist_t1_t2 <- iterative_clustering(data = t1_t2, parallel = FALSE)
klist_t2_t3 <- iterative_clustering(data = t2_t3, parallel = FALSE)
result_t1_t2 <- stabilitas(klist_t1_t2, common = "_0_")
result_t2_t3 <- stabilitas(klist_t2_t3, common = "_0_")
results <- st_previz(results = list(result_t1_t2, result_t2_t3),
                     times = c("t1_t2", "t2_t3"))
st_scatter(data = results, order = "median",
           times = c("t1_t2", "t2_t3"), grid_lines = TRUE,
           side_scale = 0.2)
```

Index