Package: MCseqReplic (via r-universe)

July 2, 2026

Version 1.1.0

Date 2026-07-01

Title Monte Carlo Simulations of Time Changes in Sequences

Depends R (>= 3.0.0), TraMineR (>= 2.2-2), ggplot2

Imports WeightedCluster (>= 1.6.0), aricode (>= 1.1.0), doParallel, parallelly, foreach, doSNOW, stats, vegan, wCorr, Rdpack, TraMineRextras (>= 0.6.8), dplyr, tidyr, magrittr

RdMacros Rdpack

VignetteBuilder knitr

Suggests knitr, rmarkdown

Description Generates replicated sets of sequences with Monte Carlo simulated timing changes and computes various indicators for evaluating effects of timing uncertainty on sequence analysis results. See Ritschard, G. and Liao, T.F. (2026): ``Assessing the Impact of Timing Errors in Sequence Analysis''. International Journal of Social Research Methodology <doi:10.1080/13645579.2026.2666297>.

License GPL (>= 2)

URL `https://traminer.unige.ch`

Encoding UTF-8

Maintainer Gilbert Ritschard <`gilbert.ritschard@unige.ch`>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Gilbert Ritschard [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0001-7776-0903`>), Tim F. Liao [ctb] (ORCID: <`https://orcid.org/0000-0002-1296-7660`>)

Repository https://cran.r-universe.dev

Date/Publication 2026-07-02 01:20:02 UTC

RemoteUrl https://github.com/cran/MCseqReplic

RemoteRef HEAD

RemoteSha 3ceb564116cd47bd22d0fa42bd1e4adf08faf3e6

Contents

`MCclustcomp` *Comparing MC-clusters with cluster of observed data*

Description

Cluster comparison indices (CCI) between clusters based on observed data and each of the MC-replicated partitions.

Usage

```
MCclustcomp(clustlist, clust.o = NULL, weights = NULL, AMI = FALSE)
```

Arguments

`clustlist`	List of MC-replicated vectors of cluster memberships.
`clust.o`	Cluster memberships based on observed dissimilarities.
`weights`	vector of doubles. Case weights. If `NULL` (default), equal weights are used.
`AMI`	logical. AMI is more costly! Should AMI also be computed. Deafult is FALSE.

Details

When `diss.o=NULL`, the last element of `disslist` is taken as `diss.o` and the other elements as sets of MC-replicated dissimilarities.

For a description of the CCIs, see (Sundqvist et al. 2022).

Value

A table with in columns the list of comparison scores provided by `aricode::compare_clustering` for each replicated set, except Chi2, which is replaced by Cramer's V.

References

Chiquet J, Rigaill G, Sundqvist M, Dervieux V, Bersani F (2023). "aricode: Efficient Computations of Standard Clustering Comparison Measures." Comprehensive R Archive Network, CRAN. doi:10.32614/CRAN.package.aricode.

Sundqvist M, Chiquet J, Rigalli G (2022). "Adjusting the adjusted Rand Index: A multinomial story." *Computational Statistics*, **38**(1), 327-347. doi:10.1007/s00180022012307.

See Also

`compare_clustering`

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
set.seed(25)
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list, method="LCS")
diss.o <- seqdist(s.exdata, method="LCS")
## cluster per MC-dissimilarity matrices
library(WeightedCluster)
clust.o <- wcKMedoids(diss.o, k=2, cluster.only=TRUE)
clustlist <- lapply(disslist, wcKMedoids, k=2, cluster.only=TRUE)
res <- MCclustcomp(clustlist, clust.o=clust.o)
res
```

`MCclustqual` *Cluster quality measures by MC-sets*

Description

Cluster quality measures for a range of number of groups by MC-replicated set.

`ggplotMCcqi` makes a ggplot of the range of values of the selected CQI by MC-sets and for the observed sequences. When `attr(data,"obs")` is `TRUE`, the range of CQI values for the observed sequences is also plotted.

The print method only prints by default the `qual.max` and `max.freq` tables of MCclustQ objects.

Usage

```
MCclustqual(
  disslist,
  ncluster = 10,
  clustmeth = "PAM",
  weights = NULL,
  core = 1,
  snow = TRUE,
  verbose = !silent,
  silent = FALSE,
  ...
)

ggplotMCcqi(
  data,
  cqi = "PBC",
  meancqi = TRUE,
  scalelwd = 1,
  linecolor = NULL,
  ...
)

## S3 method for class 'MCclustQ'
print(x, all = FALSE, nMC = 5, ...)
```

Arguments

`disslist`	List of MC-dissimilarity matrices (or `dist` objects).
`ncluster`	integer vector. Maximum number of groups. Default is `10`. CQIs are computed for the range `2:ncluster`
`clustmeth`	character. Clustering method. Either `"PAM"` (default) or a `stats::hclust` method.
`weights`	vector of doubles. Case weights. If `NULL` (default), equal weights are used.

`core`	Integer or `"auto"`. Number of cores for parallel computing If `"auto"`, the maximum available cores are used.
`snow`	Logical. If `TRUE`, `doSNOW` is used for parallel computing, otherwise `doParallel` is used.
`verbose`	Logical. Should waiting and timing messages be printed?
`silent`	Logical. Deprecated, use `!verbose` instead!
`...`	further arguments passed to or from other methods.
`data`	an `MCclustQ` object as returned by `MCclustqual`
`cqi`	string. The name of the selected CQI.
`meancqi`	logical. Should the range of mean values of the selected CQI be plotted?
`scalelwd`	double. Line width scale value.
`linecolor`	vector of three line colors in the order Mean, Obs, MCset. If `NULL`, default colors are used and if of length less than 3, default colors are used for the first elements.
`x`	`MCclustQ` object as returned by `MCclustqual`.
`all`	logical. Should tables by MC-sets also be printed? Default is `FALSE`.
`nMC`	numeric. Maximal number of MC-sets for which optimal size by CQIs are printed. Default is 5.

Details

When `attr(MCdisslist,"obs")` is `TRUE`, the last element of `disslist` is treated as the dissimilarity matrix of the observed sequences.

`MCclustqual` computes the range of CQI values for all the CQIs included in the `stats` element returned by `WeightedCluster::wcClusterQuality`.

Value

List of length 3:
- `qual.tab`: list of tables of cluster quality statistics per MC-dissimilarity matrix,
- `qual.max`: table of cluster number k for which the statistics reach their maximum (minimum for HC) by MC-sets and observed sequence set (rows),
- `max.freq`: the frequency table of optimal k over the MC-replicated sets, and

`ggplotMCcqi` returns the ggplot object.

the print method returns the last printed tables.

Author(s)

Gilbert Ritschard

References

Studer M (2013). “WeightedCluster Library Manual: A practical guide to creating typologies of trajectories in the social sciences with R.” LIVES Working Papers 24, NCCR LIVES, Switzerland. doi:10.12682/lives.22961658.2013.24.

See Also

as.clustrange, wcKMedRange

MCclustqual.

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
set.seed(25)
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list, method="LCS")
diss.o <- seqdist(s.exdata, method="LCS")
## cluster per MC-dissimilarity matrices
res <- MCclustqual(disslist,ncluster=3, verbose=FALSE)
res
ggplotMCcqi(res,"PBC")
```

`MCcompgrp` *Group comparison by MC-sets*

Description

Collects statistics and p-values for the comparison of groups of sequences.

Usage

```
MCcompgrp(
  disslist,
  group,
  weights = NULL,
  dissassoc.args = list(),
  dissCompare.args = list(),
  verbose = TRUE
)
```

Arguments

`disslist`	list of dissimilarity matrices or `dist` objects.
`group`	vector of group memberships of length equal to number of rows of the dissimilarity matrices.
`weights`	vector of case weights
`dissassoc.args`	list of additional arguments passed to `TraMineR::dissassoc`.
`dissCompare.args`	list of additional arguments passed to `TraMineRextras::dissCompare`.
`verbose`	logical. Should messages be printed?

Details

The function collects the values of R2 and its p-value (Studer et al. 2011) returned by `TraMineR::dissassoc` and the values of LRT, its p-value, and delta BIC (Liao and Fasang 2020) returned by `TraMineRextras::dissCompare`. Since `dissCompare` works only with two groups, only R2 and its p-value are returned when there are more than two groups.

Except for `group` and `weights`, `dissassoc` and `dissCompare` are called by default with the default values of their arguments. This can be changed by passing the wanted arguments as a list to `dissassoc.args` and `dissCompare.args`.

The R2 and its p-value are computed by `dissassoc`, which computes th p-value using permutation tests. The default number of permutation is `R=1000` but this can be changed by means of the `dissassoc.args` argument, for example, by passing `dissassoc.args = list(R=500)`.

The LRT and delta BIC are computed by `dissCompare`, which computes the LRT for samples of `s` data, with `s` possibly greater than the number of observed data. When `s=0` (default in `MCcompgrp`), no sampling is applied. `dissCompare` computes the p-value of LRT using the appropriate Chi-square distribution. In case of multiple samples, i.e. when `s` is smaller than the greatest group size, `BFopt=1` is used by default. `BFopt=NULL` could generate unpredictable results in that case.

References

Liao TF, Fasang AE (2020). "Comparing Groups of Life-Course Sequences Using the Bayesian Information Criterion and the Likelihood-Ratio Test." *Sociological Methodology*, **51**(1), 44-85. doi:10.1177/0081175020959401.

Ritschard G, Liao TF (2026). "Assessing the Impact of Timing Errors in Sequence Analysis." *International Journal of Social Research Methodology*. doi:10.1080/13645579.2026.2666297.

Studer M, Ritschard G, Gabadinho A, Müller NS (2011). "Discrepancy Analysis of State Sequences." *Sociological Methods and Research*, **40**(3), 471-510. doi:10.1177/0049124111415372.

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="t1 t2 t3 t4 sex
        a a b b f
        a a b b f
        b b a a f
```

```
                a c c b m
                b b a c m
                b b a c m
                ", header=TRUE)
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata[,1:4], weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
set.seed(25)
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list, method="LCS")
## Group comparison per MC-dissimilarity matrices
res <- MCcompgrp(disslist,group=exdata$sex)
res
```

`MCdisscorr` *Correlation between observed and MC-simulated distances*

Description

Correlation between observed and MC-simulated distances

Usage

```
MCdisscorr(disslist, diss.o = NULL, method = "Spearman", weights = NULL)
```

Arguments

`disslist`	List of matrices or dist objects: the MC-replicated dissimilarities
`diss.o`	Matrix or dist object: Observed dissimilarities
`method`	String. One of `"Spearman"` (default) and `"Pearson"`.
`weights`	vector of doubles. Case weights. If `NULL` (default), equal weights are used.

Details

When `diss.o=NULL`, the last element of `disslist` is taken as `diss.o` and the other elements as sets of MC-replicated dissimilarities.

Value

vector of correlation between observed and MC-dissimilarities.

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
set.seed(25)
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3, include.obs=TRUE)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list)
MCdisscorr(disslist)
```

`MCdisslist` *List of dissimilarity matrices*

Description

Compute the dissimilarity matrix for each of the provided sets of sequences.

Usage

```
MCdisslist(
  MCrseqdata,
  method = "LCS",
  seqref = NULL,
  full.matrix = FALSE,
  use.udiss = FALSE,
  ...
)
```

Arguments

`MCrseqdata`	List of state sequence objects of class `stslist`.
`method`	string. Name of a distance method (see `seqdist`).
`seqref`	state sequence object of class `stslist`. Fixed reference sequences.
`full.matrix`	logical. Should pairwise distances be returned in matrix form? If `FALSE` (default), a list of `dist` objects is returned. Applies only when `seqref=NULL`.
`use.udiss`	logical. Should computation be based on unique sequences?
`...`	further arguments passed to `seqdist`.

Details

When `use.udiss=TRUE`, the function first computes dissimilarities between unique merged replicated sequences through a single call to `seqdist()` and the set of dissimilarity matrices are then extracted from the resulting distance matrix. This is generally faster when the number of unique merged replicated sequences is less than `sqrt(number of replicated datasets) * (sample size)`, which can be checked with `MCnunique`.

Value

list of dissimilarity matrices or `dist` objects with logical attribute `"obs"`, which is `TRUE` when the list includes the dissimilarities between observed sequences as last element.

See Also

`MCseqReplicate`, `MCudist` and examples in their help pages.

`MCExtractDist` *Extract k-th dissimilarity matrix from u.diss*

Description

Extract k-th dissimilarity matrix from u.diss

Usage

```
MCExtractDist(u.diss, k, full.matrix = FALSE)
```

Arguments

`u.diss`	`u.diss` object returned by `MCudist`: dissimilarities between unique replicated sequences.
`k`	integer. Subset index number for which the dissimilarity matrix must be extracted
`full.matrix`	logical. If `FALSE`, the distance matrix is returned as a `dist` object. Ignored for distances to reference sequences.

Value

a dissimilarity matrix or distance object.

See Also

`MCudist`

`MCmdscorr` *Correlation between 1st MDS factor of observed and MC-simulated distances*

Description

Correlation between 1st MDS factor of observed and MC-simulated distances

Usage

```
MCmdscorr(
  disslist,
  diss.o = NULL,
  method = "Spearman",
  weights = NULL,
  what = "corr",
  core = 1,
  snow = TRUE,
  verbose = !silent,
  silent = FALSE
)
```

Arguments

`disslist`	List of matrices or dist objects: the MC-replicated dissimilarities
`diss.o`	Matrix or dist object: Observed dissimilarities
`method`	String. One of `"Spearman"` (default) and `"Pearson"`.
`weights`	vector of doubles. Case weights. If `NULL` (default), equal weights are used.
`what`	String. One of `"corr"` (correlations, default), `"mds"` (list of mds scores), and `"both"`.
`core`	Integer or `"auto"`. Number of cores for parallel computing If `"auto"`, the maximum available cores are used.
`snow`	Logical. If `TRUE`, `doSNOW` is used for parallel computing, otherwise `doParallel` is used.
`verbose`	Logical. Should waiting and timing messages be printed?
`silent`	Logical. Deprecated, use `!verbose` instead.

Details

When `diss.o=NULL`, the last element of `disslist` is taken as `diss.o` and the other elements as sets of MC-replicated dissimilarities.

Value

when `what="corr"`, vector of correlation between mds of dissimilarities in MC-replicated sets, when `what="mds"`, of first mds scores, and when `what="both"`, list with `corr` as first element and `mdslist`, the list of mds scores as second element.

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
set.seed(25)
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list)
MCmdscorr(disslist)
```

`MCnunique` *Number of unique replicated sequences*

Description

Number of unique replicated sequences

Usage

```
MCnunique(MCrseqdata, check = FALSE)
```

Arguments

`MCrseqdata`	list of replicated `stslist` state sequence datasets (all of same size and with same alphabet.
`check`	logical. When `TRUE`, check if the number of unique replicated sequences is less than `sqrt(number of replicated datasets) * (sample size)`?

Value

nu number of unique replicated sequences and, when `check=TRUE`, `u.ok` the check result.

See Also

`MCudist`, `MCseqdistSE`

`MCpj` *Generate distribution of timing errors*

Description

Generates a distribution of timing errors that complies with the provided expected size of non-zero timing errors and the expected probability of no error.

Usage

```
MCpj(Emean, pzero = NULL, maxterr = 10, pinterv = 0.99, fill.short.side = TRUE)
```

Arguments

`Emean`	scalar or vector of size two. Expected size of non-zero timing errors. If a vector, the first value is used for negative errors and the second value for positive errors. If a scalar, the value is used for both negative and positive errors. Values must be strictly greater than 1.
`pzero`	number in range [0,1]. Probability of no-error. If `NULL` (default), `pzero` is set to the the greatest probability of zero between the right and left side Poisson distributions.
`maxterr`	integer. Maximal error size to consider. Default is 10.
`pinterv`	control value used for solving numerically an implicit function. Default is .99 and should be increased in case the zero of the implicit function cannot be found because of ending values of same sign.
`fill.short.side`	logical. Should the shortest side be filled with zeros to equal length of the other side. Default is `TRUE`.

Details

Currently `MCseqReplicate` expects a vector Pj with same number of backward and forward error values. To comply with this, the shorter side of Pj is by default filled with zeros.

Value

The vector of probabilities Pj with the computed `lambda` values as attribute.

See Also

`MCseqReplicate`

Examples

```
# expected timing error of 1.2 on each side
MCpj(Emean=1.2, pzero=.4)

# expected backward timing error higher than for forward errors
MCpj(Emean=c(3.5,1.2), pzero=.4)
```

`MCratios` *Ratios of distances on their standard errors*

Description

Ratios of the observed distances to their MC standard errors and of the mean MC-simulated distances to the standard error of the mean.

Usage

```
MCratios(object, diss.o = NULL)
```

Arguments

`object`	Object of class `distMC` as generated by `MCseqdistSE`.
`diss.o`	Matrix or `dist` object. Pairwise dissimilarities between observed sequences.

Details

The standard error of the mean simulated distances is `mean.se` = `MC.se/sqrt(R)` (or `mean.se` = `MC.sd/R` when object is obtained with `seqdistMCSE::seqdistMCSE`, because there are `R*R` simulated distances in that case). The ratios computed are `diss.z` = `diss.o/MC.se`, where `diss.o` is the distance between observed sequences, and `MC.mean.z` = `MC.mean/mean.se` with `MC.mean` the mean of the MC-simulated distances.

When `diss.o=NULL`, the `diss.o` element of `object` is used when it exists.

This function is handy to get afterwards ratios for outcome of `MCseqdistSE` obtained with `ratios=FALSE`.

Value

`diss.z`, `MC.mean.z`, and `mean.se` (the three as `dist` objects).

Author(s)

Gilbert Ritschard

See Also

`print.MCratios` and `MCseqdistSE`

`MCseqdistSE` *Distance standard errors derived from sets of MC-replicated sequences*

Description

Computes the mean and standard deviation of each element of the pairwise distance matrix across sets of MC-replicated sequences.

Usage

```
MCseqdistSE(
  dissrepl = "LCS",
  MCrseqdata = NULL,
  udiss = FALSE,
  full.matrix = FALSE,
  ...
)
```

Arguments

`dissrepl`	list, string, or object of class `u.diss`. If a list, list of same length as `MCrseqdata`. List of dissimilarity matrices or `dist` objects. If a character string, a method name for computing the dissimilarities with `MCudist`. Can also be an object of class `u.diss` previously computed with `MCudist`.
`MCrseqdata`	list of MC-replicated sequence datasets of class `stslist`. The last element is supposed to be the observed dataset.
`udiss`	logical. When `dissrepl` is a distance method, should distance be computed with `MCudist`. See details.
`full.matrix`	logical. Should dissimilarities be organized in matrix form? Default is `FALSE` in which case dissimilarity matrices are converted into `dist` objects. If `TRUE`, dissimilarity `dist` objects are converted into matrices.
`...`	further arguments passed to `MCudist` or `MCdisslist` when `dissrepl` is a method name.

Details

Providing `u.diss` distances or computing distances with `MCudist` may be faster and can save space when the number of unique replicated sequences is smaller than the sample size times the squared root of R, which can be checked with `MCnunique`. When the number of unique replicated sequences largely exceeds the threshold, it is more efficient to compute distance matrices separately for each updated set of sequences with `MCdisslist` or by setting `udiss=FALSE`.

Value

Five objects:
`MCmean` Mean of distance objects over replicated sets of sequences.
`MCsd` Standard deviation of distances over replicated sets of sequences.
In addition, when the observed distances are provided as last element of the `dissrepl` list:
`MCbias` Difference between observed distance and `MCmean`
`MCse` Standard error of individual distances.
`MCmse` Mean square error of individual distances.
The five objects are of class `dist` when `attr(MCrseqdata,"toref")==FALSE` and matrices otherwise.

See Also

`MCseqReplicate`, `MCdisslist`, `MCudist`, `print.distMC`, `summary.distMC`

Examples

```
# example code
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 MC-replicated sequence datasets
altseq.list <- MCseqReplicate(s.exdata, J=1, R=3, include.obs=TRUE)
## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list, method="HAM")

MCdselist <- MCseqdistSE(disslist)
print(MCdselist)

MCratioslist <- MCratios(MCdselist)
print(MCratioslist)
```

`MCseqReplicate` *Generate R altered sequence data sets.*

Description

R `stslist` state sequence objects are generated by applying the chosen timing error model to the provided state sequence object.

Usage

```
MCseqReplicate(
  seqdata,
  J = 1,
  R = 20,
  silent = FALSE,
  unique = FALSE,
  model = "keep.dss",
  jfixed = FALSE,
  kchanges = NULL,
  include.obs = FALSE
)
```

Arguments

`seqdata`	A state sequence `stslist` object as generated by `seqdef`.
`J`	Integer or vector of positive numbers. If an integer, maximal timing error (number of unit periods around first state of new spell. Default is `J=1`. If a vector, weights of the timing errors `k = -K, -(K-1), ..., K-1, K`, where `2K + 1` is the length of `J`. The vector length must be odd.
`R`	Integer. Number of random replicated sequence data. Default is `R=10`. The weights are internally normalized to sum to 1.
`silent`	Logical. Should waiting and timing messages be hidden?
`unique`	Logical. Should only unique sequences be replicated? Default is `FALSE`. If `TRUE` weights will reflect the multiple occurrences of each original unique sequence.
`model`	String. Time alteration model. One of `"keep.dss"` (default), `"indep"` (suppress spells erased by move of transition), and `"relative"` (keep time until next transitions unchanged).
`jfixed`	Logical. Should same error j be applied to all transitions in a sequence? Default is `FALSE`.
`kchanges`	Integer, string, or `NULL`. If integer, number of transitions whose time can potentially be altered in each sequence. If `"rand"`, the number of potential changes is randomly selected for each sequence. If `NULL` (default), all transitions can potentially be altered.
`include.obs`	logical. Should the observed sequence data be added as last element.

Details

This function is handy for testing how outcome of a sequence analysis may vary with timing errors in the reported sequences.

Use the vector form of `J` to specify the probability distribution of the timing error. See function `MCpj` to generate a probability vector that complies with expected mean timing errors.

Value

List of R altered `stslist` objects plus observed sequence object as last element when `include.obs=TRUE`.

Author(s)

Gilbert Ritschard

References

Ritschard G, Liao TF (2026). “Assessing the Impact of Timing Errors in Sequence Analysis.” *International Journal of Social Research Methodology*. doi:10.1080/13645579.2026.2666297.

See Also

`MCpj`

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
(altseq.list <- MCseqReplicate(s.exdata, J=1, R=3))

## list of dissimilarity matrices
suppressMessages(dist.list <- lapply(altseq.list, seqdist, method="LCS", full.matrix=FALSE))
dist.list

## Can also be obtained with MCdisslist, which offers option use.udiss;
## use.udiss=TRUE is faster when number of unique merged replicated
## sequences is less than n*sqrt(R).
suppressMessages(dist.list <- MCdisslist(altseq.list, method="LCS", use.udiss=TRUE))

## Replication based on expected left and right non-zero errors of 1.1
##  and assuming a 0.5 probability of no error
Pj <- MCpj(Emean=1.1, pzero=.5)
(altseq2.list <- MCseqReplicate(s.exdata, J=Pj, R=3))
```

`MCudist` *Dissimilarities between unique replicated sequences*

Description

Returns the dissimilarity matrix (or `dist` object) between merged replicated sequences with the disaggregation indexes as attribute.

Usage

```
MCudist(MCrseqdata, method = "LCS", seqref = NULL, ...)
```

Arguments

`MCrseqdata`	list of replicated `stslist` state sequence datasets (all of same size and with same alphabet)
`method`	string. Name of distance method (see `seqdist`).
`seqref`	state sequence object of class `stslist`. Fixed reference sequences.
`...`	Further arguments passed to `seqdist`

Value

object of class `u.diss` (pairwise dissimilarities between unique sequences) with two attributes: `sdx`, inverted aggregation indexes, `N`, number of datasets, and `obs`, logical indicating whether `k=N` corresponds to observed sequences.

See Also

`MCExtractDist`

Examples

```
## mini test data, 6 sequences of length 4, 4 unique sequences
exdata <- read.table(text="
                a a b b
                a a b b
                b b a a
                a c c b
                b b a c
                b b a c
                ")
weights=rep(1, nrow(exdata))
s.exdata <- seqdef(exdata, weights = weights, id=paste("id",1:nrow(exdata), sep=""))

## 3 altered sequence datasets
(altseq.list <- MCseqReplicate(s.exdata, J=1, R=3))

MCnunique(altseq.list, check=TRUE)


u.diss <- MCudist(altseq.list, method="LCS", full.matrix=FALSE)
## Dissimilarities within first MC-set
MCExtractDist(u.diss, 1)

## list of dissimilarity matrices
disslist <- MCdisslist(altseq.list, use.udiss=TRUE)
```

`print.distMC` *Print method for distMC objects*

Description

Prints, for each pair of the first `n` sequences, the mean and/or the standard deviation of the MC-replicated distances between sequences. When available, ratios are also printed by default.

Usage

```
## S3 method for class 'distMC'
print(x, n = 6, what = "all", ...)
```

Arguments

`x`	`distMC` object as returned by `MCseqdistSE`.
`n`	Integer. Number of first sequences. Default is 6. If `n==0` or there are less than `n` sequences, results are printed for all pairs of sequences.
`what`	character string. One of `"mean"`, `"sd"`, `"bias"`, `"both"`, and `"all"` (default). When `"all"`, ratios, when present are printed together with the mean and standard deviation. When `"both"`, means and standard deviations are printed.
`...`	further arguments passed to or from other methods.

Value

Last printed table, a `matrix` when `toref` attribute is `TRUE` and a `dist` object otherwise.

Author(s)

Gilbert Ritschard

See Also

`MCseqdistSE`, `summary.distMC`.

`print.MCratios` *Print method for MCratios objects*

Description

Prints ratios for each pair of the first `n` sequences.

Usage

```
## S3 method for class 'MCratios'
print(x, n = 6, what = "all", ...)
```

Arguments

`x`	`MCratios` object as returned by `MCratios`.
`n`	Integer. Number of first sequences. Default is 6. If `n==0` or there are less than `n` sequences, results are printed for all pairs of sequences.
`what`	character string. One of `"all"` (default), `"diss"`, `"mean"`, and `"se"` .
`...`	further arguments passed to or from other methods.

Value

Last printed table, a `matrix` when `toref` attribute is `TRUE` and a `dist` object otherwise.

Author(s)

Gilbert Ritschard

See Also

`MCratios`.

`summary.distMC` *Summary method for distMC objects*

Description

Prints summary statistics of the observed dissimilarity `diss`, the mean `MC.mean`, standard deviation `MC.sd`, and standard error of dissimilarities between MC-replicated sequences, and the ratios `diss/MC.se` and `MC.mean/MC.se`. Reported statistics concern all distances between original sequences.

Usage

```
## S3 method for class 'distMC'
summary(object, ..., silent = FALSE)
```

Arguments

`object`	`distMC` object as returned by `MCseqdistSE`.
`...`	further arguments passed to or from other methods.
`silent`	logical: Should additional info be displayed?

Value

`fivenumb` table with the statistics (min, Q1, med, Q3, max) of the observed dissimilarities, the mean, standard deviation, and standard error of the MC-simulated dissimilarities, standardized ratios, MC-bias and mean squared errors when available.

Author(s)

Gilbert Ritschard

See Also

`MCseqdistSE`, `print.distMC`

`summary.MCratios` *Summary method for MCratios objects*

Description

Prints summary statistics of the ratios `diss/MC.se` and `MC.mean/MC.se`. Reported statistics concern all distances between original sequences.

Usage

```
## S3 method for class 'MCratios'
summary(object, ..., weights = NULL, silent = FALSE, thresh = 2)
```

Arguments

`object`	`MCratios` object as returned by `MCratios`.
`...`	further arguments passed to or from other methods.
`weights`	vector of doubles. Case weights.
`silent`	logical: Should additional info be displayed?
`thresh`	real: threshold for counting ratios less than `thresh`

Value

`fivenumb` table with the statistics (min, Q1, med, Q3, max) of `mean.se` and the standardized ratios `diss.z` and `MC.mean.z`.

Author(s)

Gilbert Ritschard

See Also

`MCseqdistSE`, `print.distMC`

Index