

Package ‘MCseqReplic’

April 29, 2026

Version 1.0.0

Date 2026-04-29

Title Monte Carlo Simulations of Time Changes in Sequences

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

Imports WeightedCluster (>= 1.6.0), aricode, doParallel, foreach,
doSNOW, iterators, stats, vegan, wCorr

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](https://doi.org/10.1080/13645579.2026.2666297)>.

License GPL (>= 2)

URL <http://traminer.unige.ch>

Encoding UTF-8

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

RoxygenNote 7.3.2

NeedsCompilation no

Author Gilbert Ritschard [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-7776-0903>>)

Repository CRAN

Date/Publication 2026-04-29 19:00:13 UTC

Contents

MCclustcomp	2
MCclustqual	3
MCdisscorr	5
MCdisslist	6
MCExtractDist	7
MCmdscorr	7
MCnunique	9

MCpj	10
MCratios	11
MCseqdistSE	12
MCseqReplicate	13
MCudist	15
print.distMC	17
print.MCratios	17
seqdistMCSE	18
summary.distMC	21
summary.MCratios	22

Index	23
--------------	-----------

MCclustcomp	<i>Comparing MC-clusters with cluster of observed data</i>
-------------	--

Description

Comparison indexes between clusters based on observed data and each of MC-replicated clusters.

Usage

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

Arguments

<code>clustlist</code>	List of MC-replicated vectors of cluster memberships.
<code>clust.o</code>	Cluster memberships based on observed dissimilarities.
<code>weights</code>	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

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

See Also

[clustComp](#)

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 <- MCdislist(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.

Usage

```

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

```

Arguments

<code>dissslist</code>	List of MC-dissimilarity matrices (or <code>dist</code> objects).
<code>ncluster</code>	integer vector. Range of number of groups. Default is 2:10.
<code>clustmeth</code>	character. Clustering method. Either "PAM" (default) or a <code>stats::hclust</code> method.
<code>weights</code>	vector of doubles. Case weights. If NULL (default), equal weights are used.
<code>core</code>	Integer. Number of cores for parallel computing.
<code>snow</code>	Logical. If TRUE, <code>doSNOW</code> is used for parallel computing, otherwise <code>doParallel</code> is used.
<code>silent</code>	Logical. Should waiting and timing messages be hidden?
<code>...</code>	Further arguments passed to clustering functions.

Details

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

Value

A list with two lists: `qual.tab`, list of tables of cluster quality statistics per MC-dissimilarity matrix, and `qual.max` list of cluster number `k` for which the statistics reach their maximum (minimum for HC), `max.freq`, the frequency table of maximum over the MC-replicated sets, and `qual.obs`, cluster quality indexes for the observed sequences.

See Also

[as.clustrange](#), [wckMedRange](#)

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
dissslist <- MCdissslist(altseq.list, method="LCS")
diss.o <- seqdist(s.exdata, method="LCS")
```

```
## cluster per MC-dissimilarity matrices
res <- MCclustqual(disslist,ncluster=3)
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(
  MCcrseqdata,
  method = "LCS",
  seqref = NULL,
  full.matrix = FALSE,
  use.udiss = FALSE,
  ...
)
```

Arguments

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

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 $\sqrt{\text{number of replicated datasets} \times \text{sample size}}$, 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	<i>Extract k-th dissimilarity matrix from u.diss</i>
---------------	--

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	<i>Correlation between 1st MDS factor of observed and MC-simulated distances</i>
-----------	--

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,
  silent = FALSE
)
```

Arguments

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

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
dissslist <- MCdissslist(altseq.list)
MCmdscorr(dissslist)

```

MCnunique	<i>Number of unique replicated sequences</i>
-----------	--

Description

Number of unique replicated sequences

Usage

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

Arguments

MCrseqdata	list of replicated stslst 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 $\sqrt{\text{number of replicated datasets} * (\text{sample size})}$?

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

<code>Emean</code>	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.
<code>pzero</code>	number in range [0,1]. Probability of no-error. If NULL (default), <code>pzero</code> is set to the the greatest probability of zero between the right and left side Poisson distributions.
<code>maxterr</code>	integer. Maximal error size to consider. Default is 10.
<code>pinterv</code>	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.
<code>fill.short.side</code>	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` and `seqdistMCSE`.
`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`, because there are $R \times R$ 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 `seqdistMCSE` obtained with `ratios=FALSE`.

Value

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

Author(s)

Gilbert Ritschard

See Also

`MCseqdistSE` and `print.MCratios`.

MCseqdistSE	<i>Distance standard errors derived from sets of MC-replicated sequences</i>
-------------	--

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 <code>u.diss</code> . If a list, list of same length as <code>MCrseqdata</code> . List of dissimilarity matrices or <code>dist</code> objects. If a character string, a method name for computing the dissimilarities with <code>MCudist</code> . Can also be an object of class <code>u.diss</code> previously computed with <code>MCudist</code> .
MCrseqdata	list of MC-replicated sequence datasets of class <code>stslst</code> . The last element is supposed to be the observed dataset.
udiss	logical. When <code>dissrepl</code> is a distance method, should distance be computed with <code>MCudist</code> . See details.
full.matrix	logical. Should dissimilarities be organized in matrix form? Default is <code>FALSE</code> in which case dissimilarity matrices are converted into <code>dist</code> objects. If <code>TRUE</code> , dissimilarity <code>dist</code> objects are converted into matrices.
...	additional arguments passed to <code>MCudist</code> or <code>MCdisslist</code> when <code>dissrepl</code> 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 `disssrepl` 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](#), [MCdissslist](#), [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
dissslist <- MCdissslist(altseq.list, method="HAM")

MCdselist <- MCseqdistSE(dissslist)
print(MCdselist)

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

MCseqReplicate

Generate R altered sequence data sets.

Description

R `stslst` 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

<code>seqdata</code>	A state sequence <code>stslst</code> object as generated by <code>seqdef</code> .
<code>J</code>	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 $k = -K, -(K-1), \dots, K-1, K$, where $2K + 1$ is the length of <code>J</code> . The vector length must be odd.
<code>R</code>	Integer. Number of random replicated sequence data. Default is $R=10$. The weights are internally normalized to sum to 1.
<code>silent</code>	Logical. Should waiting and timing messages be hidden?
<code>unique</code>	Logical. Should only unique sequences be replicated? Default is <code>FALSE</code> . If <code>TRUE</code> weights will reflect the multiple occurrences of each original unique sequence.
<code>model</code>	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).
<code>jfixed</code>	Logical. Should same error <code>j</code> be applied to all transitions in a sequence? Default is <code>FALSE</code> .
<code>kchanges</code>	Integer, string, or <code>NULL</code> . 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 <code>NULL</code> (default), all transitions can potentially be altered.
<code>include.obs</code>	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 `stslst` objects plus observed sequence object as last element when `include.obs=TRUE`.

Author(s)

Gilbert Ritschard

References

Ritschard, G. and Liao, T.F. (2026). Assessing the Impact of Timing Errors in Sequence Analysis. *International Journal of Social Research Methodology*. Forthcoming

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 stslst 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 stslst. 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

[seqdistMCSE](#), [MCratios](#).

seqdistMCSE	<i>Mean and standard deviation of dissimilarities between pairs of randomly altered sequences.</i>
-------------	--

Description

For each pair of sequences, returns the mean and standard deviation (MCSE) of the dissimilarities between all combinations of MC-replicated sequences, where sequences are replicated with random timing changes.

Usage

```
seqdistMCSE(
  seqdata,
  method = "LCS",
  J = 1,
  R = 50,
  replic = "by.pair",
  verbose = TRUE,
  core = 1,
  unique = TRUE,
  model = "keep.dss",
  jfixed = FALSE,
  kchanges = NULL,
  ratios = TRUE,
  snow = TRUE,
  ...
)
```

Arguments

seqdata	A state sequence <code>stslst</code> object as generated by <code>seqdef</code> .
method	Character string. Dissimilarity measure to compute distances. Default is "LCS". See <code>seqdist</code> .
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 $k = -K, -(K-1), \dots, K-1, K$, where $2K + 1$ is the length of J. The vector length must be odd.
R	Integer. Number of random replications of each sequence. Default is R=50.
replic	Character string. One of "by.pair" (default) and "once". When "by.pair" a fresh set of replications of the sequences is generated for each distance. When "once", replications of each sequence are generated once and reused for each distance involving the sequence.
verbose	Logical. Should waiting and timing messages be printed?
core	Integer. Number of cores to use for parallel computation.
unique	Logical. Should simulations for distances between identical pairs of sequences be run only once? Default is TRUE.
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. If "rand", the number of potential changes is randomly selected for each sequence. If NULL (default), all transitions can potentially be altered.
ratios	Logical. Should standardized ratios and the standard error of mean simulated distances be returned? Default is TRUE.
snow	Logical. If TRUE, <code>doSNOW</code> is used for parallel computing, otherwise <code>doParallel</code> is used.
...	Further arguments passed to <code>seqdist</code> .

Details

Let B_x be the set of R sequences derived from a sequence x by randomly altering the timing of the transitions (state changes) in x . The MC standard error of the dissimilarity $d(x, y)$ between two sequences x and y is the empirical standard deviation of the dissimilarities between the sequences of B_x and those of B_y . There are R^2 such MC-simulated dissimilarities for each pair of observed sequences.

By default, MC standard errors are computed for distances between unique sequences and results are then expanded to all sequences. In addition, results for pairs of identical sequences are expanded to all such pairs in `seqdata`. With `unique=FALSE`, the computation is redone for each identical pairs and, therefore, results can vary across such identical pairs. Setting `unique=TRUE` (default) can save much computation time when same sequences occur multiple times.

A progress bar is displayed when verbose=TRUE. However, the progress bar works only with option snow=TRUE for parallel computing.

seqdistMCSE is much slower than [MCseqdistSE](#), which considers only distances within sets of replicated sequences (generated with [MCseqReplicate](#)) instead of all combinations of replicated sequences.

Value

A list of class `distMC` with for each pairwise distance:

- `MC.mean` (`dist` object) MC means of distances between MC-replicated sequences,
 - `MC.se` (`dist` object) MC standard deviations of distances between MC-replicated sequences,
 - `args.dist` list of arguments passed to `seqdist`,
 - `diss.o` (`dist` object) observed distances between sequences,
- and when `ratios = TRUE`:
- `diss.z` (`dist` object) ratios `diss.u/MC.se`,
 - `MC.mean.z` (`dist` object) ratios `MC.mean/mean.se`,
 - `mean.se` (`dist` object) standard errors of `MC.mean`.

Author(s)

Gilbert Ritschard

References

Liao, T.F. and G. Ritschard (2023). Evaluating uncertainty of dissimilarity measures between state sequences. Manuscript in preparation.

See Also

[MCseqdistSE](#), [print.distMC](#), [summary.distMC](#), and [MCRatios](#)

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=""))

## Here we call function seqdistMCSE
MCd <- seqdistMCSE(s.exdata, method="LCS", J=1, R=50, core=1, verbose=TRUE)

## Results for distances between first sequences
MCd
```

```
## Summary statistics refer to all distances between original sequences
summary(MCd)
```

summary.distMC	<i>Summary method for distMC objects</i>
----------------	--

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

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

Value

five-nb 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

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

Value

five numeric 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

`as.clustrange`, 4

`clustComp`, 2

`hclust`, 4

`MCclustcomp`, 2

`MCclustqual`, 3

`MCdisscorr`, 5

`MCdislist`, 6, 12, 13

`MCExtractDist`, 7, 16

`MCmdscorr`, 7

`MCnunique`, 6, 9, 12

`MCpj`, 10, 14, 15

`MCratios`, 11, 18, 20

`MCseqdistSE`, 9, 11, 12, 17, 20–22

`MCseqReplicate`, 7, 10, 13, 13, 20

`MCudist`, 7, 9, 12, 13, 15

`print.distMC`, 13, 17, 20–22

`print.MCratios`, 11, 17

`seqdef`, 14, 19

`seqdist`, 6, 19

`seqdistMCSE`, 11, 18, 18

`summary.distMC`, 13, 17, 20, 21

`summary.MCratios`, 22

`wcKMedRange`, 4