

Package ‘iCAMP’

January 9, 2021

Type Package

Title Infer Community Assembly Mechanisms by Phylogenetic-Bin-Based Null Model Analysis

Version 1.3.4

Date 2021-1-8

Author Daliang Ning

Maintainer Daliang Ning <ningdaliang@ou.edu>

Imports vegan,parallel,permute,ape,bigmemory,nortest,minpack.lm,Hmisc,stats4

Depends R (>= 3.5)

Description To implement a general framework to quantitatively infer Community Assembly Mechanisms by Phylogenetic-bin-based null model analysis, abbreviated as ‘iCAMP’ (Ning et al 2020) <doi:10.1038/s41467-020-18560-z>. It can quantitatively assess the relative importance of different community assembly processes, such as selection, dispersal, and drift, for both communities and each phylogenetic group (‘bin’). Each bin usually consists of different taxa from a family or an order. The package also provides functions to implement some other published methods, including neutral taxa percentage (Burns et al 2016) <doi:10.1038/ismej.2015.142> based on neutral theory model (Sloan et al 2006) <doi:10.1111/j.1462-2920.2005.00956.x> and quantifying assembly processes based on entire-community null models (‘QPEN’, Stegen et al 2013) <doi:10.1038/ismej.2013.93>. It also includes some handy functions, particularly for big datasets, such as phylogenetic and taxonomic null model analysis at both community and bin levels, between-taxa niche difference and phylogenetic distance calculation, phylogenetic signal test within phylogenetic groups, midpoint root of big trees, etc. Version 1.3.x mainly improved the function for ‘QPEN’ and added function ‘icamp.cate()’ to summarize ‘iCAMP’ results for different categories of taxa (e.g. core versus rare taxa).

License GPL-2

URL <https://github.com/DaliangNing/iCAMP1>

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-09 06:30:16 UTC

R topics documented:

icAMP-package	3
bmntd	4
bmntd.big	6
bmpd	8
bNRI.bin.big	9
bNRIn.p	13
bNTI.big	16
bNTI.bin.big	18
bNTIn.p	22
change.sigindex	25
cohend	26
dist.3col	28
dist.bin.3col	29
dniche	30
example.data	32
icamp.big	34
icamp.bins	40
icamp.boot	41
icamp.cate	43
icamp.out	45
match.2col	56
match.name	57
maxbigm	59
midpoint.root.big	60
mntdn	62
mpdn	63
NRI.p	64
NTI.p	67
null.norm	69
pdist.big	71
ps.bin	73
qp.bin.js	76
qpen	78
RC.bin.bigc	81
RC.pc	83
snm	86
taxa.binphy.big	89
tree.droot	92
tree.path	93

iCAMP-package

Infer Community Assembly Mechanisms by Phylogenetic-bin-based null model analysis

Description

This package is to implement a general framework to quantitatively infer Community Assembly Mechanisms by Phylogenetic-bin-based null model analysis, abbreviated as iCAMP (Ning et al 2020). It can quantitatively assess the relative importance of different community assembly processes, such as selection, dispersal, and drift, for both communities and each phylogenetic group ('bin'). Each bin usually consists of different taxa from a family or an order. The package also provides functions to implement some other published methods, including neutral taxa percentage (Burns et al 2016) based on neutral theory model (Sloan et al 2006) and quantifying assembly processes based on entire-community null models (Stegen et al 2013). It also includes quite a few handy functions, particularly for big datasets, such as phylogenetic and taxonomic null model analysis at both community and bin levels, between-taxa niche difference and phylogenetic distance calculation, phylogenetic signal test within phylogenetic groups, midpoint root of big trees, etc. URL: <https://github.com/DaliangNing/iCAMP1>

Version 1.2.4: the first formal version of iCAMP for CRAN. Version 1.2.5: correct typo in description and fix the error of memory.limit issue. Version 1.2.6: revise the help document of `qpen` to include an example for big datasets. Version 1.2.7: remove `setwd` in functions; add options to specify file names; change `dontrun` to `donttest` and revise `save.wd` in some help documents. Version 1.2.8: revise `dniche` to avoid unnecessary file. Version 1.2.9: update iCAMP paper newly published on Nature Communications and the GitHub link. Version 1.2.10: fix minor bug when `output.wd` is NULL in `icamp.big`. Version 1.2.11: fix minor bug in `icamp.big` when `comm` is `data.frame`. Version 1.3.1: add `bNTI.big` and `bMNTD.big`, and revise `qpen` to handle big datasets better. Version 1.3.2: revise `icamp.bins` to fix error when an input taxonomy name has unrecognizable character; revise `icamp.boot` to fix error when there is no outlier. Version 1.3.3: add `icamp.cate` to summary for each category of taxa, e.g. core versus rare taxa. Version 1.3.4: typo and format.

Details

Package: iCAMP
Type: Package
Version: 1.3.4
Date: 2021-1-8
License: GPL-2

Author(s)

Daliang Ning <ningdaliang@ou.edu>

References

- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Burns, A.R., Stephens, W.Z., Stagaman, K., Wong, S., Rawls, J.F., Guillemin, K. et al. (2016). Contribution of neutral processes to the assembly of gut microbial communities in the zebrafish over host development. *Isme Journal*, 10, 655-664.
- Sloan, W.T., Lunn, M., Woodcock, S., Head, I.M., Nee, S. & Curtis, T.P. (2006). Quantifying the roles of immigration and chance in shaping prokaryote community structure. *Environmental Microbiology*, 8, 732-740.
- Stegen, J.C., Lin, X., Fredrickson, J.K., Chen, X., Kennedy, D.W., Murray, C.J. et al. (2013). Quantifying community assembly processes and identifying features that impose them. *Isme Journal*, 7, 2069-2079.

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree
# since need to save some outputs to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
pd.wd=paste0(save.wd,"/pdbg")
nworker=4 # parallel computing thread number
rand.time=20 # usually use 1000 for real data.

bin.size.limit=5 # for real data, usually use a proper number
# according to phylogenetic signal test or try some settings
# then choose the reasonable stochasticity level.
# our experience is 12, or 24, or 48.
# but for this example dataset which is too small, have to use 5.

setwd(save.wd)
icamp.out=icamp.big(comm=comm,tree=tree,pd.wd=pd.wd,
                    rand=rand.time, nworker=nworker,
                    bin.size.limit=bin.size.limit)
```

 bmntd

beta mean nearest taxon distance (betaMNTD)

Description

Calculates beta MNTD (beta mean nearest taxon distance, Webb et al 2008) for taxa in each pair of communities in a given community matrix.

Usage

```
bmntd(comm, pd, abundance.weighted = TRUE,
       exclude.conspecific = FALSE, time.output=FALSE,
       unit.sum=NULL)
```

Arguments

<code>comm</code>	matrix or data.frame, community data matrix, rownames are sample names, colnames are OTU ids.
<code>pd</code>	matrix, pairwise phylogenetic distance matrix.
<code>abundance.weighted</code>	logic, whether weighted by species abundance, default is TRUE, means weighted.
<code>exclude.conspecific</code>	logic, whether conspecific taxa in different communities be exclude from beta MNTD calculations, default is FALSE.
<code>time.output</code>	logic, whether to count calculation time, default is FALSE.
<code>unit.sum</code>	NULL or a number or a numeric vector. When unit.sum is not NULL and a beta diversity index is calculated for a bin, the taxa abundances will be divided by unit.sum to calculate the relative abundances. usually, unit.sum can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this special transformation.

Details

beta mean nearest taxon distance for taxa in each pair of communities. Modified from 'comdistnt' in package 'picante' (Kembel et al 2010), this function includes matrix multiplication to be efficient for medium size dataset.

Value

result is a distance object of pairwise beta MNTD between samples.

Note

Version 3: 2020.8.16, add examples. Version 2: 2018.10.15, add unit.sum option. if unit.sum!=NULL, will calculate relative abundance according to unit.sum. Version 1: 2015.9.23

Author(s)

Daliang Ning

References

Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.

Kembel, S.W., Cowan, P.D., Helmus, M.R., Cornwell, W.K., Morlon, H., Ackerly, D.D. et al. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463-1464.

See Also[bNTIn.p](#)**Examples**

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
bmntd.wt=bmntd(comm, pd, abundance.weighted = TRUE,
               exclude.conspecifics = FALSE)
```

bmntd.big*beta mean nearest taxon distance (betaMNTD) from big data*

Description

Calculates beta MNTD (beta mean nearest taxon distance, Webb et al 2008) for taxa in each pair of communities in a given community matrix, using bigmemory (Kane et al 2013) to deal with too large dataset.

Usage

```
bmntd.big(comm, pd.desc = "pd.desc", pd.spname, pd.wd,
           spname.check = FALSE, abundance.weighted = TRUE,
           exclude.conspecifics = FALSE, time.output = FALSE)
```

Arguments

<code>comm</code>	matrix or data.frame, community data matrix, rownames are sample names, colnames are taxa ids.
<code>pd.desc</code>	character, the name to describe bigmemory file of phylogenetic distance matrix, default is "pd.desc".
<code>pd.spname</code>	vector, the OTU ids (species names) in exactly the same order as the phylogenetic matrix rows or columns
<code>pd.wd</code>	the path of the folder saving the phylogenetic distance matrix.
<code>spname.check</code>	logic, whether to check the OTU ids (species names) in community matrix and phylogenetic distance matrix are the same.
<code>abundance.weighted</code>	logic, whether weighted by species abundance, default is TRUE, means weighted.
<code>exclude.conspecifics</code>	logic, whether conspecific taxa in different communities be exclude from beta MNTD calculations, default is FALSE.
<code>time.output</code>	logic, whether to count calculation time, default is FALSE.

Details

beta mean nearest taxon distance for taxa in each pair of communities. Improved from 'comdistnt' in package 'picante' (Kembel et al 2010). This function adds bigmemory part (Kane et al 2013) to deal with large dataset.

Value

result is a distance object.

Note

Version 4: 2020.12.5, copy from package NST to iCAMP to improve the function qpen. Version 3: 2020.9.9, remove setwd; change dontrun to donttest and revise save.wd in help doc. Version 2: 2020.8.22, add to NST package, update help document. Version 1: 2017.3.13

Author(s)

Daliang Ning (ningdaliang@ou.edu)

References

Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.

Kembel, S.W., Cowan, P.D., Helmus, M.R., Cornwell, W.K., Morlon, H., Ackerly, D.D. et al. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463-1464.

Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree

# since it needs to save some file to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to use.
nworker=2 # parallel computing thread number
pd.big=iCAMP::pdist.big(tree = tree,wd = save.wd, nworker = nworker)
bmntd.wt=bmntd.big(comm=comm, pd.desc = pd.big$pd.file,
                  pd.spname = pd.big$tip.label, pd.wd = pd.big$pd.wd,
                  abundance.weighted = TRUE)
```

 bmpd

Beta mean pairwise distance (betaMPD)

Description

Calculates mean pairwise distance separating taxa in each pair of communities in a given community matrix.

Usage

```
bmpd(comm, pd, abundance.weighted = TRUE, na.zero = TRUE,
      time.output = FALSE, unit.sum = NULL)
```

Arguments

comm	matrix or data.frame, community data matrix, rownames are sample names, colnames are OTU ids.
pd	matrix, pairwise phylogenetic distance matrix.
abundance.weighted	logic, whether weighted by species abundance, default is TRUE, means weighted.
na.zero	logic. when the sum of a row (a sample) is zero in community data matrix, the relative abundance will be NAN. Sometimes, to avoid some problem in following calculation, this kind of NAN value need be set as zero. Default is TRUE.
time.output	logic, whether to count calculation time, default is FALSE.
unit.sum	When a beta diversity index is calculated for a bin, the taxa abundances will be divided by unit.sum to calculate the relative abundances. usually, unit.sum can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this special transformation.

Details

beta mean pairwise distance.

Value

Output is a distance object of pairwise betaMPD between samples.

Note

Version 3: 2020.8.16, add examples. Version 2: 2018.10.3, add unit.sum option. if unit.sum!=NULL, will calculate relative abundance according to unit.sum Version 1: 2015.8.21.

Author(s)

Daliang Ning

References

Webb CO, Ackerly DD, and Kembel SW. 2008. Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics* 18:2098-2100

See Also

[bNRIn.p](#)

Examples

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
bmpd.wt=bmpd(comm, pd, abundance.weighted = TRUE)
```

bNRI.bin.big	<i>Calculate beta net relatedness index (betaNRI) for each phylogenetic bin</i>
--------------	---

Description

Perform null model test based on a phylogenetic beta diversity index, beta mean pairwise distance (betaMPD), in each bin; calculate beta net relatedness index (betaNRI. Webb et al 2008), or modified Raup-Crick metric, or confidence level based on the comparison between observed and null betaMPD in each bin. The package bigmemory (Kane et al 2013) is used to handle very large phylogenetic distance matrix.

Usage

```
bNRI.bin.big(comm, pd.desc, pd.spname, pd.wd, pdid.bin, sp.bin,
             spname.check = FALSE, nworker = 4, memo.size.GB = 50,
             weighted = c(TRUE, FALSE), rand = 1000, output.bMPD = FALSE,
             sig.index=c("SES", "Confidence", "RC", "bNRI"),
             unit.sum = NULL, correct.special = FALSE,
             detail.null=FALSE, special.method=c("MPD", "MNTD", "both"),
             ses.cut=1.96, rc.cut=0.95, conf.cut=0.975)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a species or OTU or gene, thus rownames should be sample IDs, col-names should be taxa IDs.
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.

<code>pd.wd</code>	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.
<code>pdid.bin</code>	list, each element is a vector of integer, indicating which rows/columns in the big phylogenetic matrix represent the taxa in a bin.
<code>sp.bin</code>	one-column matrix, rownames are taxa IDs (i.e. OTU IDs), the only column shows the bin ID of each taxon. Bin IDs are integers in the same order as the elements in the list of <code>pdid.bin</code> .
<code>spname.check</code>	logic, whether to check the OTU ids (species names) in community matrix and phylogenetic distance matrix are the same.
<code>nworker</code>	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
<code>memo.size.GB</code>	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
<code>weighted</code>	Logic, consider abundances or not (just presence/absence). default is TRUE.
<code>rand</code>	integer, randomization times. default is 1000.
<code>output.bMPD</code>	logic, if TRUE, the output will include beta mean pairwise distance (betaMPD).
<code>sig.index</code>	character, the index for null model significance test. SES or bNRI, standard effect size, i.e. beta net relatedness index (betaNRI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on betaMPD, i.e. count the number of null betaMPD lower than observed betaMPD plus a half of the number of null betaMPD equal to observed betaMPD, to get alpha, then calculate betaMPD-based RC as $(2 \times \alpha - 1)$. default is SES. If input a vector, only the first element will be used.
<code>unit.sum</code>	NULL or a number or a numeric vector. When a beta diversity index is calculated for a bin, the taxa abundances will be divided by <code>unit.sum</code> to calculate the relative abundances. Usually, <code>unit.sum</code> can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this transformation.
<code>correct.special</code>	logic, whether to correct the special cases. Default is FALSE.
<code>detail.null</code>	logic, if TRUE, the output will include all the null values. Default is FALSE.
<code>special.method</code>	When <code>correct.special</code> is TRUE, which method will be used to check underestimation of deterministic pattern(s) in special cases. MPD, use null model test based on mean pairwise distance; MNTD, use null model test of mean nearest taxon distance; both, use null model test of both MPD and MNTD. Default is MPD.
<code>ses.cut</code>	numeric, the cutoff of significant standard effect size, default is 1.96.
<code>rc.cut</code>	numeric, the cutoff of significant modified Raup-Crick metric, default is 0.95.
<code>conf.cut</code>	numeric, the cutoff of significant one-side confidence level, default is 0.975.

Details

The beta net relatedness index (betaNRI; Webb et al. 2008, Stegen et al 2012) is calculated for each phylogenetic bin. betaNRI is a standardized measure of the mean pairwise distance between samples/communities (betaMPD). Parallel computing is used to improve the speed.

The null model algorithm is "taxa shuffle" (Kembel 2009), i.e. shuffling taxa labels across the tips of the phylogenetic tree to randomize phylogenetic relationships among species. In this function, taxa will be randomized across all bins.

In the betaNRI of each bin, the diagonal are set as zero. If the randomized results are all the same, the standard deviation will be zero and betaNRI will be NAN. In this case, betaNRI will be set as zero, since the observed result is not differentiable from randomized results.

Modified RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed beta diversity index deviates from null expectation, which could be a better metric than standardized effect size (betaNRI) in some cases, e.g. null values do not follow normal distribution.

Value

Output is a list with following elements:

index	list, each element is a square matrix of betaNRI (or RC or Confidence based on betaMPD) values of a bin. The elements (bins) are in the same order as in the input pdid.bin.
sig.index	character, indicates the index for null model significance test, SES (i.e. betaNRI), RC, or Confidence.
betaMPD.obs	Output only if output.bMPD is TRUE. A list, each element is a square matrix of observed beta MPD values of a bin. The elements (bins) are in the same order as in the input pdid.bin.
rand	Output only if detail.null is TRUE. A list, each element is a matrix with null values of beta MPD for each turnover of a bin. The elements (bins) are in the same order as in the input pdid.bin.
special.crct	Output only if detail.null is TRUE. NULL if correct.special is FALSE. A list with three elements, corresponding to three different null model significance testing indexes, i.e. SES, RC, and Confidence. Each element is a matrix, where the value is zero if the result for a turnover of a bin does not need to correct, otherwise there will be a corrected value.

Note

Version 6: 2020.9.1, remove setwd. change dontrun to donttest and revise save.wd in help doc.
 Version 5: 2020.8.18, update help document, add example. Version 4: 2020.8.1, change RC option to sig.index, add detail.null and conf.cut. Version 3: 2018.10.15, add unit.sum, correct.special.
 Version 2: 2016.3.26, add RC option. Version 1: 2015.12.16

Author(s)

Daliang Ning

References

- Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.
- Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.
- Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *Isme Journal*, 6, 1653-1664.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.
- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[bNRIn.p,bmpd](#)

Examples

```
# this function is usually used in icamp.big when setting phylo.rand.scale="across",
# means randomization across all bins in phylogenetic null model.
data("example.data")
comm=example.data$comm
tree=example.data$tree
pdid.bin=example.data$pdid.bin
sp.bin=example.data$sp.bin

# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the example on your computer after change the path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)
rand.time=20 # usually use 1000 for real data.

bNRIBins=bNRI.bin.big(comm=comm, pd.desc=pd.big$pd.file, pd.spname=pd.big$tip.label,
                      pd.wd=pd.big$pd.wd, pdid.bin=pdid.bin, sp.bin=sp.bin,
                      spname.check = FALSE, nworker = nworker, memo.size.GB = 50,
                      weighted = TRUE, rand = rand.time, output.bMPD = FALSE,
                      sig.index="SES",unit.sum = NULL, correct.special = TRUE,
                      detail.null=FALSE, special.method="MPD")
```

bNRIn.p

*Calculate beta net relatedness index with parallel computing***Description**

Perform null model test based on a phylogenetic beta diversity index, beta mean pairwise distance (betaMPD); calculate beta net relatedness index (betaNRI), or modified Raup-Crick metric, or confidence level based on the comparison between observed and null betaMPD. Run by parallel computing.

Usage

```
bNRIn.p(comm, dis, nworker = 4, memo.size.GB = 50,
         weighted = c(TRUE, FALSE), rand = 1000, output.bMPD = c(FALSE, TRUE),
         sig.index = c("SES", "Confidence", "RC", "bNRI"),
         unit.sum = NULL, correct.special = FALSE, detail.null = FALSE,
         special.method = c("MPD", "MNTD", "both"),
         ses.cut = 1.96, rc.cut = 0.95, conf.cut = 0.975)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a species or OTU or gene, thus rownames should be sample IDs, col-names should be taxa IDs.
dis	matrix, pairwise phylogenetic distance matrix.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memo.size.GB	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
weighted	logic, whether to use abundance-weighted or unweighted metrics. Default is TRUE.
rand	integer, randomization times. default is 1000.
output.bMPD	logic, if TRUE, the output will include beta mean pairwise distance (betaMPD).
sig.index	character, the index for null model significance test. SES or bNRI, standard effect size, i.e. beta net relatedness index (betaNRI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on betaMPD, i.e. count the number of null betaMPD lower than observed betaMPD plus a half of the number of null betaMPD equal to observed betaMPD, to get alpha, then calculate betaMPD-based RC as $(2 \times \alpha - 1)$. default is SES. If input a vector, only the first element will be used.
unit.sum	NULL or a number or a numeric vector. When a beta diversity index is calculated for a bin, the taxa abundances will be divided by unit.sum to calculate the relative abundances. Usually, unit.sum can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this transformation.

<code>correct.special</code>	logic, whether to correct the special cases. Default is FALSE.
<code>detail.null</code>	logic, if TRUE, the output will include all the null values. Default is FALSE.
<code>special.method</code>	When <code>correct.special</code> is TRUE, which method will be used to check underestimation of deterministic pattern(s) in special cases. MPD, use null model test based on mean pairwise distance; MNTD, use null model test of mean nearest taxon distance; both, use null model test of both MPD and MNTD. Default is MPD.
<code>ses.cut</code>	numeric, the cutoff of significant standard effect size, default is 1.96.
<code>rc.cut</code>	numeric, the cutoff of significant modified Raup-Crick metric, default is 0.95.
<code>conf.cut</code>	numeric, the cutoff of significant one-side confidence level, default is 0.975.

Details

The beta net relatedness index (betaNRI; Webb et al. 2008, Stegen et al. 2012) is a standardized measure of the mean pairwise distance between samples/communities (betaMPD). Parallel computing is used to improve the speed.

The null model algorithm is "taxa shuffle" (Kembel 2009), i.e. shuffling taxa labels across the tips of the phylogenetic tree to randomize phylogenetic relationships among species.

In the output of beta NRI, the diagonal are set as zero. If the randomized results are all the same, the standard deviation will be zero and betaNRI will be NAN. In this case, beta NRI will be set as zero, since the observed result is not differentiable from randomized results.

Modified RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed beta diversity index deviates from null expectation, which could be a better metric than standardized effect size (betaNRI) in some cases, e.g. null values do not follow normal distribution.

Value

Output is a list with following elements:

<code>index</code>	a square matrix of betaNRI (or RC or Confidence based on betaMPD) values.
<code>sig.index</code>	character, indicates the index for null model significance test, SES (i.e. betaNRI), RC, or Confidence.
<code>betaMPD.obs</code>	Output only if <code>output.bMPD</code> is TRUE. A square matrix of observed beta MPD values.
<code>rand</code>	Output only if <code>detail.null</code> is TRUE. A matrix with null values of beta MPD for each turnover.
<code>special.crct</code>	Output only if <code>detail.null</code> is TRUE. it will be NULL if <code>correct.special</code> is FALSE. Otherwise, it will be a list with three elements, corresponding to three different null model significance testing indexes, i.e. SES, RC, and Confidence. Each element is a square matrix, where the value is zero if the result for a turnover does not need to correct, otherwise there will be a corrected value.

Note

Version 4: 2020.8.18, update help document, add example. Version 3: 2020.8.1, change RC opiton to sig.index, add detail.null and conf.cut. Version 2: 2018.10.3, correct special cases Version 1: 2016.3.26.

Author(s)

Daliang Ning

References

Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.

Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.

Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *Isme Journal*, 6, 1653-1664.

Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[bmpd](#)

Examples

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
nworker=2 # parallel computing thread number
rand.time=4 # usually use 1000 for real data.
bNRI=bNRIn.p(comm=comm, dis=pd, nworker = nworker, memo.size.GB = 50,
              weighted = TRUE, rand = rand.time, output.bMPD = FALSE,
              sig.index = "SES", unit.sum = NULL, correct.special = TRUE,
              detail.null = FALSE, special.method = "MPD")
```

bNTI.big

*Beta nearest taxon index (betaNTI) from big data***Description**

To calculate pairwise beta nearest taxon index (betaNTI) by randomizing in the whole species pool or within each group. Package bigmemory (Kane et al 2013) is used to deal with large datasets.

Usage

```
bNTI.big(comm, meta.group=NULL, pd.desc="pd.desc",
         pd.spname,pd.wd, spname.check=TRUE,
         nworker=4, memo.size.GB=50, weighted=TRUE,
         exclude.consp=FALSE,rand=1000,output.dtail=FALSE,
         RC=FALSE, trace=TRUE)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each colname is a species or OTU or gene, thus rownames should be sample IDs, colnames should be taxa IDs.
meta.group	matrix or data.frame, a one-column (n x 1) matrix indicating which metacommunity each sample belongs to. rownames are sample IDs. first column is metacommunity IDs. Such that different samples can belong to different metacommunities. If input a n x m matrix, only the first column is used. NULL means all samples belong to the same metacommunity. Default is NULL, means all samples are under the same metacommunity (the same regional species pool).
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.
spname.check	logic, whether to check the OTU ids (species names) in community matrix and phylogenetic distance matrix are the same.
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memo.size.GB	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
weighted	Logic, consider abundances or not (just presence/absence). default is TRUE.
exclude.consp	Logic, should conspecific taxa in different communities be exclude from MNTD calculations? default is FALSE. The same as in the function bmntd.
rand	integer, randomization times. default is 1000.

output.detail	logic, if TRUE, the betaNTI, RC value, observed betaMNTD, all null betaMNTD values will all be output, if FALSE, only output betaNTI or RC.
RC	logic, whether to use modified RC metrics to evaluate significance of betaMNTD instead of betaNTI (standardized effect size).
trace	logic, whether to show the progress when the code is running.

Details

The beta nearest taxon index (betaNTI) is a standardized measure of the mean phylogenetic distance to the nearest taxon between samples/communities (betaMNTD) and quantifies the extent of terminal clustering, independent of deep level clustering. There are a lot of null models for randomization, but this function only use phylogeny shuffle (the same as taxa.labels in ses.mntd).

In the output of betaNTI, the diagonal are set as zero. If the randomized results are all the same, the standard deviation will be zero and betaNTI will be NAN. In this case, beta NTI will be set as zero, since the observed result is not differentiable from randomized results. If the observed betaMNTD has NA values, the corresponding betaNTI will remain NA. Modified RC (Chase 2010) is another metric to evaluate how the observed betaMNTD deviates from null expectation, which could be a better metric than standardized effect size (classic betaNTI) in some cases.

Value

If output.detail=FALSE (default), a matrix of betaNTI values (if RC=FALSE) or RC values (if RC=TRUE) is returned. If output.detail=TRUE, a list is returned.

bNTI: a matrix of pairwise betaNTI values; RC.bMNTD: a matrix of RC values based on null model test of betaMNTD; bMNTD: observed betaMNTD values; bMNTD.rand: a matrix of all null results;

Note

Version 2: 2020.12.5, included into iCAMP package to improve the function qpen. Version 1: 2017.7.12

Author(s)

Daliang Ning

References

- Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.
- Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.
- Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *Isme Journal*, 6, 1653-1664.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.

Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[bmntd.big,qpen](#)

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree

# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer after change the path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)
rand.time=20 # usually use 1000 for real data.
bNTI=bNTI.big(comm=comm, pd.desc=pd.big$pd.file,
              pd.sname=pd.big$tip.label,pd.wd=pd.big$pd.wd,
              sname.check=TRUE, nworker=nworker, memo.size.GB=50,
              weighted=TRUE, exclude.consp=FALSE,rand=rand.time,
              output.dtail=FALSE, RC=FALSE, trace=TRUE)
```

bNTI.bin.big	<i>Calculate beta nearest taxon index (betaNTI) for each phylogenetic bin</i>
--------------	---

Description

Perform null model test based on a phylogenetic beta diversity index, beta mean phylogenetic distance to the nearest taxon (betaMNTD), in each bin; calculate beta nearest taxon index (betaNTI), or modified Raup-Crick metric, or confidence level based on the comparison between observed and null betaMNTD in each bin. The package bigmemory (Kane et al 2013) is used to handle very large phylogenetic distance matrix.

Usage

```
bNTI.bin.big(comm, pd.desc, pd.sname, pd.wd, pdid.bin,
             sp.bin, sname.check = FALSE, nworker = 4,
             memo.size.GB = 50, weighted = c(TRUE, FALSE),
             rand = 1000, output.bMNTD = c(FALSE, TRUE),
             sig.index=c("SES", "Confidence", "RC", "bNTI"),
             unit.sum = NULL, correct.special = FALSE, detail.null = FALSE,
             special.method = c("MNTD", "MPD", "both"),
```

```
ses.cut = 1.96, rc.cut = 0.95, conf.cut = 0.975,
exclude.conspecific = FALSE)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a species or OTU or gene, thus rownames should be sample IDs, col-names should be taxa IDs.
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.
pdid.bin	list, each element is a vector of integer, indicating which rows/columns in the big phylogenetic matrix represent the taxa in a bin.
sp.bin	one-column matrix, rownames are taxa IDs (i.e. OTU IDs), the only column shows the bin ID of each taxon. Bin IDs are integers in the same order as the elements in the list of pdid.bin.
spname.check	logic, whether to check the OTU ids (species names) in community matrix and phylogenetic distance matrix are the same.
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memo.size.GB	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
weighted	Logic, consider abundances or not (just presence/absence). default is TRUE.
rand	integer, randomization times. default is 1000.
output.bMNTD	logic, if TRUE, the output will include betaMNTD.
sig.index	character, the index for null model significance test. SES or bNTI, standard effect size, i.e. beta nearest taxon index (betaNTI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on betaMNTD, i.e. count the number of null betaMNTD lower than observed betaMNTD plus a half of the number of null betaMNTD equal to observed betaMNTD, to get alpha, then calculate betaMNTD-based RC as $(2 \times \alpha - 1)$. default is SES. If input a vector, only the first element will be used.
unit.sum	NULL or a number or a numeric vector. When a beta diversity index is calculated for a bin, the taxa abundances will be divided by unit.sum to calculate the relative abundances. Usually, unit.sum can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this transformation.
correct.special	logic, whether to correct the special cases. Default is FALSE.
detail.null	logic, if TRUE, the output will include all the null values. Default is FALSE.

<code>special.method</code>	When <code>correct.special</code> is TRUE, which method will be used to check underestimation of deterministic pattern(s) in special cases. MNTD, use null model test of mean distance to the nearest taxon; MPD, use null model test based on mean pairwise distance; both, use null model test of both MPD and MNTD. Default is MNTD.
<code>ses.cut</code>	numeric, the cutoff of significant standard effect size, default is 1.96.
<code>rc.cut</code>	numeric, the cutoff of significant modified Raup-Crick metric, default is 0.95.
<code>conf.cut</code>	numeric, the cutoff of significant one-side confidence level, default is 0.975.
<code>exclude.conspecific</code>	Logic, should conspecific taxa in different communities be exclude from MNTD calculations? default is FALSE. The same as in the function <code>bmntd</code> .

Details

The beta nearest taxon index (betaNTI; Webb et al. 2008, Stegen et al. 2012) is calculated for each phylogenetic bin. betaNTI is a standardized measure of the mean phylogenetic distance to the nearest taxon between samples/communities (beta MNTD) and quantifies the extent of terminal clustering, independent of deep level clustering. Parallel computing is used to improve the speed.

The null model algorithm is "taxa shuffle" (Kembel 2009), i.e. shuffling taxa labels across the tips of the phylogenetic tree to randomize phylogenetic relationships among species. In this function, taxa will be randomized across all bins.

In the betaNTI of each bin, the diagonal are set as zero. If the randomized results are all the same, the standard deviation will be zero and betaNTI will be NAN. In this case, betaNTI will be set as zero, since the observed result is not differentiable from randomized results.

Modified RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed beta diversity index deviates from null expectation, which could be a better metric than standardized effect size (betaNTI) in some cases, e.g. null values do not follow normal distribution.

Value

Output is a list with following elements:

<code>index</code>	list, each element is a square matrix of betaNTI values of a bin. The elements (bins) are in the same order as in the input <code>pdid.bin</code> .
<code>sig.index</code>	character, indicates the index for null model significance test, SES (i.e. betaNTI), RC, or Confidence.
<code>betaMNTD.obs</code>	Output only if <code>output.bMNTD</code> is TRUE. A list, each element is a square matrix of observed beta MNTD values of a bin. The elements (bins) are in the same order as in the input <code>pdid.bin</code> .
<code>rand</code>	Output only if <code>detail.null</code> is TRUE. A list, each element is a matrix with null values of beta MNTD for each turnover of a bin. The elements (bins) are in the same order as in the input <code>pdid.bin</code> .
<code>special.crct</code>	Output only if <code>detail.null</code> is TRUE. NULL if <code>correct.special</code> is FALSE. A list with three elements, corresponding to three different null model significance testing indexes, i.e. SES, RC, and Confidence. Each element is a matrix, where

the value is zero if the result for a turnover of a bin does not need to correct, otherwise there will be a corrected value.

Note

Version 7: 2020.9.1, remove setwd. change dontrun to donttest and revise save.wd in help doc.
 Version 6: 2020.8.18, update help document, add example. Version 5: 2020.8.1, change RC opiton to sig.index, add detail.null, rc.cut and conf.cut. Version 4: 2019.11.6, add exclude.conspecific.
 Version 3: 2018.10.15, add unit.sum, correct.special. Version 2: 2016.3.26, add RC option. Version 1: 2015.12.16

Author(s)

Daliang Ning

References

- Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.
- Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.
- Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *Isme Journal*, 6, 1653-1664.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.
- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[bNTIn.p,bmntd](#)

Examples

```
# function 'bNTI.bin.big' is usually used in the main function, 'icamp.big',
# when setting phylo.rand.scale="across",
# means randomization across all bins in phylogenetic null model.

data("example.data")
comm=example.data$comm
tree=example.data$tree
pdid.bin=example.data$pdid.bin
sp.bin=example.data$sp.bin

# since pdist.big need to save output to a certain folder,
```

```
# the following code is set as 'not test'.
# but you may test the code on your computer after change the path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)
rand.time=20 # usually use 1000 for real data.

bNTIbins=bNTI.bin.big(comm=comm, pd.desc=pd.big$pd.file, pd.spname=pd.big$tip.label,
  pd.wd=pd.big$pd.wd, pdid.bin=pdid.bin, sp.bin=sp.bin,
  spname.check = TRUE, nworker = nworker, memo.size.GB = 50,
  weighted = TRUE, rand = rand.time, output.bMNTD = FALSE,
  sig.index="SES", unit.sum = NULL, correct.special = TRUE,
  detail.null = FALSE, special.method = "MNTD",
  exclude.conspecifics = FALSE)
```

bNTIn.p

*Calculate beta nearest taxon index (betaNTI) with parallel computing***Description**

Perform null model test based on a phylogenetic beta diversity index, beta mean phylogenetic distance to the nearest taxon (betaMNTD); calculate beta nearest taxon index (betaNTI), or modified Raup-Crick metric, or confidence level, based on the comparison between observed and null betaMNTD. Run by parallel computing.

Usage

```
bNTIn.p(comm, dis, nworker = 4, memo.size.GB = 50,
  weighted = c(TRUE, FALSE), exclude.consp = FALSE,
  rand = 1000, output.bMNTD = c(FALSE, TRUE),
  sig.index=c("SES", "Confidence", "RC", "bNTI"),
  unit.sum = NULL, correct.special = FALSE,
  detail.null=FALSE,
  special.method=c("MNTD", "MPD", "both"),
  ses.cut=1.96, rc.cut=0.95, conf.cut=0.975)
```

Arguments

comm	community data matrix. rownames are sample names. colnames are species names
dis	Phylogenetic distance matrix.
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memo.size.GB	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.

<code>weighted</code>	Logic, consider abundances or not (just presence/absence). default is TRUE.
<code>exclude.consp</code>	Logic, should conspecific taxa in different communities be exclude from MNTD calculations? default is FALSE. The same as in the function <code>bmntd</code> .
<code>rand</code>	integer, randomization times. default is 1000.
<code>output.bMNTD</code>	logic, if TRUE, the output will include the observed betaMNTD.
<code>sig.index</code>	character, the index for null model significance test. SES or bNTI, standard effect size, i.e. beta nearest taxon index (betaNTI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on betaMNTD, i.e. count the number of null betaMNTD lower than observed betaMNTD plus a half of the number of null betaMNTD equal to observed betaMNTD, to get alpha, then calculate betaMNTD-based RC as $(2 \times \alpha - 1)$. default is SES. If input a vector, only the first element will be used.
<code>unit.sum</code>	NULL or a number or a numeric vector. When a beta diversity index is calculated for a bin, the taxa abundances will be divided by <code>unit.sum</code> to calculate the relative abundances. Usually, <code>unit.sum</code> can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this transformation.
<code>correct.special</code>	logic, whether to correct the special cases when calculating bNTI. Default is FALSE.
<code>detail.null</code>	logic, if TRUE, the output will include all the null values. Default is FALSE.
<code>special.method</code>	When <code>correct.special</code> is TRUE, which method will be used to check underestimation of deterministic pattern(s) in special cases. MNTD, use null model test of mean distance to the nearest taxon; MPD, use null model test based on mean pairwise distance; both, use null model test of both MPD and MNTD. Default is MNTD.
<code>ses.cut</code>	numeric, the cutoff of significant standard effect size, default is 1.96.
<code>rc.cut</code>	numeric, the cutoff of significant modified Raup-Crick metric, default is 0.95.
<code>conf.cut</code>	numeric, the cutoff of significant one-side confidence level, default is 0.975.

Details

The beta nearest taxon index (beta NTI; Webb et al. 2008, Stegen et al. 2012) is a standardized measure of the mean phylogenetic distance to the nearest taxon between samples/communities (beta MNTD) and quantifies the extent of terminal clustering, independent of deep level clustering. Parallel computing is used to improve the speed.

The null model algorithm is "taxa shuffle" (Kembel 2009), i.e. shuffling taxa labels across the tips of the phylogenetic tree to randomize phylogenetic relationships among species.

In the output of betaNTI, the diagonal are set as zero. If the randomized results are all the same, the standard deviation will be zero and betaNTI will be NAN. In this case, beta NTI will be set as zero, since the observed result is not differentiable from randomized results.

Modified RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed beta diversity index deviates from null expectation, which could be a better metric than standardized effect size (betaNTI) in some cases, e.g. null values do not follow normal distribution.

Value

Output is a list with following elements:

index	a square matrix of betaNTI (or RC or Confidence based on betaMNTD) values.
sig.index	character, indicates the index for null model significance test, SES (i.e. betaNTI), RC, or Confidence.
betaMNTD.obs	Output only if output.bMNTD is TRUE. A square matrix of observed beta MNTD values.
rand	Output only if detail.null is TRUE. A matrix with null values of beta MNTD for each turnover.
special.crct	Output only if detail.null is TRUE. it will be NULL if correct.special is FALSE. Otherwise, it will be a list with three elements, corresponding to three different null model significance testing indexes, i.e. SES, RC, and Confidence. Each element is a square matrix, where the value is zero if the result for a turnover does not need to correct, otherwise there will be a corrected value.

Note

Version 6: 2020.8.18, update help document, add example. Version 5: 2020.8.1, change RC option to sig.index, add detail.null, rc.cut and conf.cut. Version 4: 2018.10.15, consider special cases. Version 3: 2016.3.26, add RC option. Version 2: 2015.9.23, set diag of bNTI = 0 and set 0/0 = 0 for bNTI. Version 1: 2015.4.1

Author(s)

Daliang Ning

References

- Webb, C.O., Ackerly, D.D. & Kembel, S.W. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics*, 24, 2098-2100.
- Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.
- Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *ISME Journal*, 6, 1653-1664.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.
- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[bmntd](#)

Examples

```

data("example.data")
comm=example.data$comm
pd=example.data$pd
nworker=2 # parallel computing thread number
rand.time=4 # usually use 1000 for real data.
bNTI=bNTIn.p(comm=comm, dis=pd, nworker = nworker, memo.size.GB = 50,
             weighted = TRUE, exclude.consp = FALSE, rand = rand.time,
             output.bMNTD = FALSE, sig.index = "SES", unit.sum = NULL,
             correct.special = TRUE, detail.null = FALSE,
             special.method = "MNTD")

```

change.sigindex	<i>Change significance index option in iCAMP analysis</i>
-----------------	---

Description

This function is to change the method to calculate significance between null and observed dissimilarity and/or change the significance threshold values.

Usage

```

change.sigindex(icamp.output, sig.index = c("Confidence", "SES.RC", "SES", "RC"),
               detail.save = TRUE, detail.null = FALSE,
               ses.cut = 1.96, rc.cut = 0.95, conf.cut = 0.975)

```

Arguments

icamp.output	list, the exact output of the function icamp.big in which detail.null must be TRUE, to save all null values.
sig.index	character, Confidence means to directly count the percentage of null values higher/lower than observed value; SES.RC means to use Standard Effect Size (e.g. betaNRI, betaNTI) for phylogenetic beta diversity and use modified Raup-Crick for taxonomic beta diversity, which is typical practice in the previous method; SES means to use Standard Effect Size for both phylogenetic and taxonomic beta diversity; RC means to use modified Raup-Crick for both phylogenetic and taxonomic beta diversity.
detail.save	logic, whether to output the details, including binning information, significance indexes, bin abundances, and some key parameter settings for iCAMP analysis. Default is TRUE
detail.null	logic, whether to output all observed and null values of beta diversity indexes. Default is FALSE.
ses.cut	numeric, the cutoff of significant standard effect size, default is 1.96.
rc.cut	numeric, the cutoff of significant modified Raup-Crick index value, default is 0.95.
conf.cut	numeric, the cutoff of significant confidence level (one-tail), default is 0.975.

Details

This function is to re-calculate significance using another index or a different threshold value using previously saved null model values. Since the null values are directly extracted from previous `icamp.big` results, it can skip the most time-consuming step (randomization) and quickly complete calculation.

The default threshold values of Confidence (0.975), SES (1.96), and RC (0.95) mean to capture the 0.95 two-tail confidence level ($P=0.05$). But, SES need to assume the null values follow normal distribution. RC counts in a half of the special cases that null values are equal to observed values, which is good for obtaining a symmetric metric but theoretically has risk to misestimate significance level (but very slight). Thus, Confidence is preferred as long as the 1000-time randomization is representative.

Value

The output will be the same as `icamp.big`.

Note

Version 2: 2020.8.18, update help document, add example. Version 1: 2020.8.1

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[icamp.big](#), [null.norm](#)

Examples

```
data("icamp.out")
icamp.out.new=change.sigindex(icamp.output=icamp.out, sig.index = "Confidence")
```

cohend

Cohen's d effect size

Description

This function is to calculate the popular effect size index Cohen's d.

Usage

```
cohend(treat, control, paired = FALSE)
```

Arguments

treat	a numeric vector. treatment group.
control	a numeric vector. control group.
paired	logic. Whether the samples in treatment and control groups are paired. default is FALSE.

Details

This function computes the value of Cohen's d statistics (Cohen 1988). The effect size magnitude is performed using the thresholds proposed by Cohen (1992), i.e. $|d| < 0.2$ "negligible", $0.2 \leq |d| < 0.5$ "small", $0.5 \leq |d| < 0.8$ "medium", $|d| \geq 0.8$ "large". The variance of the d is calculate using the conversion formula reported at page 238 of Cooper et al. (2009): $((n1+n2)/(n1*n2) + .5*d^2/df) * ((n1+n2)/df)$ Its square root is output as standard deviation of d.

Value

A list of values will be returned

d	Cohen's d value, $(\text{mean}(\text{treat}) - \text{mean}(\text{control})) / \text{sd}$
sd	standard deviation of d
magnitude	a qualitative assessment of the magnitude of effect size
paired	whether the samples are paired

Note

version 1: 2016.2.12

Author(s)

Daliang Ning

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). New York: Academic Press

Cohen, J. (1992). A power primer. Psychological Bulletin, 112, 155-159.

The Handbook of Research Synthesis and Meta-Analysis (Cooper, Hedges, & Valentine, 2009)

Examples

```
x=c(1,5,8)
y=c(2,6,10)
cohend(x,y)
cohend(x,y,paired=TRUE)
```

`dist.3col`*Transform distance matrix to 3-column matrix*

Description

Transform a distance matrix to a 3-column matrix in which the first 2 columns indicate the pairwised samples/species names.

Usage

```
dist.3col(dist)
```

Arguments

`dist` a square matrix or distance object with column names and row names.

Details

In many cases, a 3-column matrix is easier to use than a distance matrix.

Value

<code>name1</code>	1st column, the first item of pairwised two items
<code>name2</code>	2nd column, the second item of pairwised two items
<code>dis</code>	3rd column, distance value of the pairwised two itmes

Note

Version 1: 2015.5.17

Author(s)

Daliang Ning

Examples

```
# In this example, dist.3col transforms the distance object  
# of Bray-Curtis dissimilarity to 3-column matrix.
```

```
data("example.data")  
comm=example.data$comm  
BC=vegan::vegdist(comm)  
BC3c=dist.3col(BC)
```

dist.bin.3col	<i>Convert a list of dist (or matrixes) to a matrix</i>
---------------	---

Description

Convert a list of distance matrixes (or square matrixes) with the same sample IDs into a matrix.

Usage

```
dist.bin.3col(dist.bin, obj.name = NULL)
```

Arguments

`dist.bin` a list, each element is a distance matrix or square matrix. all elements have exactly the same sample IDs (rownames and colnames) which are in the same order.

`obj.name` a character, as a prefix of the bin names.

Details

A tool to facilitate format transformation in iCAMP analysis.

Value

output is a matrix. The first two columns are sample IDs, and each of the following columns represent an element in the original list which usually is a bin in iCAMP analysis.

Note

Version 2: 2020.8.18, add example Version 1: 2015.8.30

Author(s)

Daliang Ning

Examples

```
# let's see a very simple example
bin.dist=as.matrix(dist(1:10))
rownames(bin.dist)<-colnames(bin.dist)<-paste0("Sample",1:10)
dist.bins=list(bin1=bin.dist,bin2=bin.dist+1,bin3=bin.dist*2)
dis.3c=dist.bin.3col(dist.bins,obj.name="test")
```

dniche *Calculate niche difference between species*

Description

Calculate niche difference between species based on each environmental variable, directly output the matrix or save the result matrix as big.matrix.

Usage

```
dniche(env, comm,
       method = c("ab.overlap", "niche.value", "prefer.overlap"),
       nworker = 4, memory.G = 50, out.dist = FALSE,
       bigmemo = TRUE, nd.wd = getwd(), nd.spname.file="nd.names.csv",
       detail.file="ND.res.rda")
```

Arguments

env	matrix or data.frame, each row is a sample, each column is an environmental factor which may be important to represent the niche, thus rownames are sample IDs, and colnames are environmental factor names.
comm	matrix or data.frame, each row is a sample, each column is a species (OTU or ASV), thus rownames are sample IDs, colnames are species/OTU/ASV IDs.
method	methods to calculate niche difference. ab.overlap means to calculate from overlap based on observed abundances along an environment gradient. niche.value means to calculate the difference from abundance weighted mean of each environment factor for each species. prefer.overlap is similar to ab.overlap, but the observed abundances of each species are divided by total abundance sum of the species before calculating overlapping. If list multiple methods as a vector, only the first element will be used.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memory.G	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
out.dist	logic, if TRUE, the output niche difference matrix of each environment factor will be a distance object, otherwise will be a matrix in the output list.
bigmemo	logic, if TRUE, big.matrix in R package bigmemory will be used to save each niche difference matrix as a big matrix on hard disk.
nd.wd	folder path, when bigmemo is TRUE, where the big matrixes are saved.
nd.spname.file	character, name of the file saving taxa IDs, which should be in exactly the same order as in the row names (and column names) of the big niche difference matrix, if bigmemo is TRUE. it should be a .csv file.
detail.file	character, name of the file saving all output information in R data format. it should be a .rda file.

Details

The method `niche.value` is to calculate niche difference as the absolute difference of niche values between each pair of species. The niche value of a species is calculated as abundance-weighted mean of each environmental factor as previously reported (Stegen et al 2012 ISME J). In the method `ab.overlap`, the abundance of each species along the gradient of an environment factor is estimated using the density function using Gaussian kernel with 512 points. Then, the niche difference between two species is calculated as the sum of absolute abundance difference at each point divided by the sum of the higher abundance at each point, like Ruzicka dissimilarity (weighted Jaccard). It is like $1 - \text{niche overlap}$ based on abundance profile overlap, thus called `ab.overlap`. The method `prefer.overlap` is very similar to `ab.overlap`, just one modification, i.e. the observed abundance of each species in each sample is divided by the total abundance of the species across all sample, to normalize the profile, before calculating niche difference.

Bigmemory (Kane et al 2013) is used to deal with large datasets.

Value

The output is a list object, with several elements.

<code>bigmemo</code>	logic, to show whether <code>big.matrix</code> is used.
<code>nd</code>	if <code>bigmemo</code> is FALSE, this is a list of matrixes or distance objects showing the niche difference matrix based on each environment factor. if <code>bigmemo</code> is TRUE, this is a list of big matrix file names.
<code>nd.wd</code>	only appear when <code>bigmemo</code> is TRUE, shows the folder path where the big matrixes are saved.
<code>names</code>	only appear when <code>bigmemo</code> is TRUE, shows species (OTU or ASV) IDs, in the same order as <code>rownames</code> and <code>colnames</code> in the niche difference matrixes.
<code>method</code>	The method used.

Note

Version 3: 2020.9.1, add `nd.spname.file` and `detail.file`; remove `setwd`; change `dontrun` to `donttest` and revise `save.wd` in help doc. Version 2: 2020.8.18, add `example`. Version 1: 2020.5.15

Author(s)

Daliang Ning

References

- Stegen, J.C., Lin, X., Konopka, A.E. & Fredrickson, J.K. (2012). Stochastic and deterministic assembly processes in subsurface microbial communities. *ISME J*, 6, 1653-1664.
- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also[ps.bin](#)**Examples**

```

data("example.data")
comm=example.data$comm
env=example.data$env

# if data is small, you do not need to use big.memory
niche.dif=dniche(env = env, comm = comm, method = "niche.value",
                nworker = 1,out.dist=FALSE,bigmemo=FALSE,nd.wd = NULL)

# if data is large, you need to use big.memory
# since big.memory need to specify a certain folder,
# it is set as 'not test'.
# but you may test the code on your computer after change the path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the big niche difference matrix.
nworker=2 # parallel computing thread number
niche.dif=dniche(env = env, comm = comm,
                method = "niche.value", nworker = nworker,
                out.dist=FALSE,bigmemo=TRUE,nd.wd = save.wd)

```

example.data

*A simple example dataset for test***Description**

A small dataset including community matrix, phylogenetic tree, treatment information, environmental factors. just for test.

Usage

```
data("example.data")
```

Format

The format is: List of 4 \$ comm : int [1:20, 1:30] 1 3 0 0 2 2 2 0 0 4- attr(*, "dimnames")=List of 2\$: chr [1:20] "s1" "s2" "s3" "s4"\$: chr [1:30] "OTU1" "OTU2" "OTU3" "OTU4" ... \$ tree :List of 4 ..\$ edge : int [1:58, 1:2] 31 32 33 34 35 36 36 35 34 37\$ edge.length: num [1:58] 0.314 0.422 0.315 0.881 0.774\$ Nnode : int 29 ..\$ tip.label : chr [1:30] "OTU1" "OTU2" "OTU3" "OTU4"- attr(*, "class")= chr "phylo" ..- attr(*, "order")= chr "cladewise" \$ treat:'data.frame': 20 obs. of 2 variables: ..\$ Management: chr [1:20] "SF" "BF" "SF" "SF"\$ Location : chr [1:20] "south" "south" "south" "south" ... \$ env :'data.frame': 20 obs. of 2 variables: ..\$ pH : num [1:20] 3.4 3.6 3.8 4 4.2 4.4 4.6 4.8 5 5.2\$ temperature: num [1:20] 12.9 4 13.7 6 8 2.7 10.4 3.2 7.6 5.8 ... \$ pd :'data.frame': 30 obs. of 30 variables: ..\$ OTU1 : num [1:30] 0


```

0.606 1.348 3.015 3.331 ... ..$ OTU2 : num [1:30] 0.606 0 1.19 2.856 3.173 ... ..$ OTU3 : num
[1:30] 1.35 1.19 0 2.05 2.37 ... ..$ OTU4 : num [1:30] 3.015 2.856 2.051 0 0.479 ... ..$ OTU5 :
num [1:30] 3.331 3.173 2.367 0.479 0 ... ..$ OTU6 : num [1:30] 2.96 2.8 2 1.9 2.22 ... ..$ OTU7
: num [1:30] 3.62 3.46 2.65 2.56 2.87 ... ..$ OTU8 : num [1:30] 4.97 4.82 4.01 3.92 4.23 ... ..$
OTU9 : num [1:30] 4.77 4.61 3.81 3.71 4.03 ... ..$ OTU10: num [1:30] 3.96 3.81 3 2.9 3.22 ... ..$
OTU11: num [1:30] 3.4 3.24 2.43 2.34 2.65 ... ..$ OTU12: num [1:30] 3.69 3.53 2.73 2.63 2.95
... ..$ OTU13: num [1:30] 5.64 5.48 4.68 4.58 4.9 ... ..$ OTU14: num [1:30] 6.1 5.94 5.14 5.04
5.36 ... ..$ OTU15: num [1:30] 4.73 4.57 3.77 3.67 3.99 ... ..$ OTU16: num [1:30] 5.77 5.61 4.8
4.71 5.03 ... ..$ OTU17: num [1:30] 5.97 5.81 5 4.91 5.22 ... ..$ OTU18: num [1:30] 5.27 5.11 4.3
4.21 4.52 ... ..$ OTU19: num [1:30] 7.56 7.4 6.6 6.5 6.82 ... ..$ OTU20: num [1:30] 7.55 7.39 6.58
6.49 6.8 ... ..$ OTU21: num [1:30] 6.49 6.33 5.53 5.43 5.75 ... ..$ OTU22: num [1:30] 6.52 6.37
5.56 5.46 5.78 ... ..$ OTU23: num [1:30] 6.68 6.52 5.72 5.62 5.94 ... ..$ OTU24: num [1:30] 6.35
6.19 5.38 5.29 5.61 ... ..$ OTU25: num [1:30] 6.37 6.21 5.41 5.31 5.63 ... ..$ OTU26: num [1:30]
5.73 5.57 4.77 4.67 4.99 ... ..$ OTU27: num [1:30] 6.23 6.07 5.27 5.17 5.49 ... ..$ OTU28: num
[1:30] 5.99 5.83 5.02 4.93 5.24 ... ..$ OTU29: num [1:30] 5.7 5.54 4.74 4.64 4.96 ... ..$ OTU30:
num [1:30] 3.94 3.78 2.97 2.88 3.19 ... $ pdid.bin:List of 3 ..$ : int [1:5] 1 2 3 4 5 ..$ : int [1:7]
6 7 8 9 10 11 12 ..$ : int [1:18] 13 14 15 16 17 18 19 20 21 22 ... $ sp.bin : 'data.frame': 30 obs.
of 1 variable: ..$ bin.id.new: num [1:30] 1 1 1 1 2 2 2 2 ... $ classification: chr [1:30, 1:6]
"Archaea" "Archaea" "Archaea" "Archaea" ... ..- attr(*, "dimnames")=List of 2 ..$ : chr [1:30]
"OTU1" "OTU2" "OTU3" "OTU4" ... ..$ : chr [1:6] "Domain" "Phylum" "Class" "Order" ...

```

Details

comm is a matrix, each row as a sample, each column as a species.

tree means phylogenetic tree.

treat is a treatment information matrix, each row as a sample, each column indicates a type of treatment.

env is a matrix of environmental factors, i.e. pH and temperature in this case.

pd is a matrix of the pairwise phylogenetic distance between species.

pdid.bin is a list, each element is a vector of integer, indicating which rows/columns in the big phylogenetic matrix represent the taxa in a bin.

sp.bin is a one-column matrix, rownames are taxa IDs (i.e. OTU IDs), the only column shows the bin ID of each taxon. Bin IDs are integers in the same order as the elements in the list of pdid.bin.

classification is a matrix to define the lineage of each taxon.

This dataset is randomly generated, just for test.

Examples

```

data(example.data)
comm=example.data$comm
tree=example.data$tree
treat=example.data$treat
env=example.data$env
pd=example.data$pd
pdid.bin=example.data$pdid.bin
sp.bin=example.data$sp.bin

```

icamp.big

*Infer community assembly mechanism by phylogenetic-bin-based null model analysis***Description**

main function of iCAMP, to perform phylogenetic-bin-based null model analysis and quantify the relative importance of different processes.

Usage

```
icamp.big(comm, tree, pd.desc = NULL, pd.spname = NULL, pd.wd = getwd(),
  rand = 1000, prefix = "iCAMP", ds = 0.2, pd.cut = NA, sp.check = TRUE,
  phylo.rand.scale = c("within.bin", "across.all", "both"),
  taxa.rand.scale = c("across.all", "within.bin", "both"),
  phylo.metric = c("bMPD", "bMNTD", "both", "bNRI", "bNTI"),
  sig.index=c("Confidence", "SES.RC", "SES", "RC"), bin.size.limit = 24,
  nworker = 4, memory.G = 50, rtree.save = FALSE, detail.save = TRUE,
  qp.save = TRUE, detail.null=FALSE, ignore.zero = TRUE,
  output.wd = getwd(), correct.special = TRUE, unit.sum = rowSums(comm),
  special.method = c("depend", "MPD", "MNTD", "both"),
  ses.cut = 1.96, rc.cut = 0.95, conf.cut=0.975,
  omit.option = c("no", "test", "omit"), meta.ab = NULL,
  treepath.file="path.rda", pd.spname.file="pd.taxon.name.csv",
  pd.backingfile="pd.bin", pd.desc.file="pd.desc")
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a taxon (a species or OTU or ASV), thus rownames should be sample IDs, colnames should be taxa IDs.
tree	phylogenetic tree, an object of class "phylo".
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function. If it is NULL, the function pd.big will be used to calculate the phylogenetic distance matrix from tree, and save it in pd.wd as a big.memory file.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.
rand	integer, randomization times. default is 1000.
prefix	character string, the prefix of those output files.
ds	numeric, the general threshold of phylogenetic distance within which the phylogenetic signal is still significant. default is 0.2.

pd.cut	numeric, the distance to the tree root where the phylogenetic tree is truncated to get strict phylogenetic bins. if pd.cut is set, the distance threshold (ds) is disabled. default is NA.
sp.check	logic, whether to match the taxa ids in community data, phylogenetic distance matrix, and tree. default is TRUE.
phylo.rand.scale	character, the scale to randomize the taxa for phylogenetic null model. "within.bin" means randomization within each bin; "across.all" means randomization across all bins; "both" means to test both methods. Default setting is within.bin.
taxa.rand.scale	character, the scale to randomize the taxa for taxonomic null model. "within.bin" means randomization within each bin; "across.all" means randomization across all bins; "both" means to test both methods. Default setting is across.all.
phylo.metric	character, the metric for phylogenetic null model analysis. bMPD (or bNRI), null model analysis based on beta mean pairwise distance (betaMPD); if sig.index is SES, it is beta net relatedness index (betaNRI). bMNTD (or bNTI), null model analysis based on beta mean nearest taxon distance (betaMNTD); if sig.index is SES, it is beta nearest taxon index (betaNTI). both, use null model test based on both bMPD and bMNTD. Default setting is based on bMPD.
sig.index	character, the index for null model significance test. Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; if set sig.index as Confidence, it will be applied to both phylogenetic and taxonomic metrics. If set as SES.RC, use standard effect size (SES) for phylogenetic metrics (i.e. betaNTI or betaNRI), and use modified Raup-Crick (RC) for taxonomic metrics (RCbray). If set as SES, use SES for both phylogenetic and taxonomic metrics. If set as RC, use RC for both phylogenetic and taxonomic metrics. default is Confidence. If input a vector, only the first one will be used.
bin.size.limit	integer, the minimal requirement of bin size (taxa numer in a bin). Default setting is 24.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memory.G	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
rtree.save	logic, whether to save the rooted tree as nwk file, if the input tree is not rooted. Default is FALSE.
detail.save	logic, whether to save the details, i.e. some key objects for iCAMP analysis, as rda file. Default is TRUE.
qp.save	logic, whether to save the relative importance of processes as csv file. Default is TRUE.
detail.null	logic, if TRUE, the output will include all the null values. Default is FALSE. But this need to be TRUE if you want to change significance testing index later using 'change.sigindex'.

ignore.zero	logic, in the community data matrix (comm), whether to remove the row(s)/column(s) of which the sum is zero. Default is TRUE.
output.wd	a folder path, where the files will be saved when rtree.save, detail.save, or qp.save is true.
correct.special	logic, whether to correct the special cases when calculating bNRI or bNTI. Default is TRUE.
unit.sum	NULL or a number or a numeric vector. When a beta diversity index is calculated for a bin, the taxa abundances will be divided by unit.sum to calculate the relative abundances, and the Bray-Cuits index in each bin will become manhattan index divided by 2. Default setting are the row sums of community matrix, which are usually sequencing depth in each sample. If set as NULL, means not to do this special transformation.
special.method	When correct.special is TRUE, which method will be used to check underestimation of deterministic pattern(s) in special cases. MPD, use null model test based on mean pairwise distance; MNTD, use null model test based on mean nearest taxon distance; depend, use MPD when phylo.metric is bMPD or bNRI, and use MNTD when phylo.metric is bMNTD or bNTI; both, use both MPD and MNTD. Default is depend
ses.cut	numeric, the cutoff of significant standard effect size, default is 1.96.
rc.cut	numeric, the cutoff of significant modified Raup-Crick index value, default is 0.95.
conf.cut	numeric, the cutoff of significant one-side confidence level, default is 0.975.
omit.option	three options about omitting small bins. "no" means to merge small bins to their nearest relatives to meet the bin size requirement, rather than omitting them; "test" means to output the information of small strict bins with a size lower than requirement, iCAMP will not be performed; "omit" means to do iCAMP analysis with strict bins which have enough species (larger than bin size requirement).
meta.ab	a numeric vector, to define the relative abundance of each species in the regional pool. Default setting is NULL, means to calculate meta.ab as average relative abundance of each species across the samples.
treepath.file	character, name of the file saving the tree.path, which is a list of all the nodes and edge lengths from root to every tip and/or node. it should be a .rda filename.
pd.spname.file	character, name of the file saving the taxa IDs, which has exactly the same order as the row names (and column names) of the big phylogenetic distance matrix. it should be a .csv filename.
pd.backingfile	character, the root name for the file for the cache of the big phylogenetic distance matrix. it should be a .bin filename.
pd.desc.file	character, name of the file to hold the backingfile description for the big phylogenetic distance matrix. it should be a .desc filename.

Details

This is the main function of iCAMP (Ning et al 2020). Most parameters can use the default settings.

To quantify various ecological processes, the observed taxa are first divided into different groups ('bins') based on their phylogenetic relationships. Then, the process governing each bin is identified based on null model analysis of the phylogenetic diversity using beta Net Relatedness Index (betaNRI), and taxonomic beta-diversities using modified Raup-Crick metric (RC; a typical setting of sig.index as SES.RC). For each bin, the fraction of pairwise comparisons with betaNRI < -1.96 is considered as the percentages of homogeneous selection, whereas those with betaNRI > +1.96 as the percentages of heterogeneous selection based on the threshold applied previously (Stegen et al 2015; Zhou and Ning 2017). Next, taxonomic diversity metric RC is used to partition the remaining pairwise comparisons with abs(NRI) <= 1.96. The fraction of pairwise comparisons with RC < -0.95 is treated as the percentages of homogenizing dispersal, while those with RC > 0.95 as dispersal limitation (Stegen et al 2013). The remains with abs(NRI) <= 1.96 and abs(RC) <= 0.95 represent the percentages of drift, diversification, weak selection and/or weak dispersal (Zhou and Ning 2017), simply designated as 'drift' (Stegen et al 2013) for convenience. The above analysis is repeated for every bin. Subsequently, the fractions of individual processes across all bins are weighted by the relative abundance of each bin, and summarized to estimate the relative importance of individual processes at the whole community level. Besides betaNRI and RC, null model significance can also be inferred by direct test based on null model distribution, which should be a preferred choice when the null model simulated values do not follow normal distribution (Veech 2012). See the references for details.

Bigmemory (Kane et al 2013) is used to deal with large datasets.

Value

If omit.option is test, the output will be a table summarizing the information of small bins.

Otherwise, the output is a list object, including one or more elements as below:

The first one or several (if set 'both' for metrics and/or randomization scale) elements are matrixes of process importances at community level. In each matrix, the first two columns will be sample ID of each turnover, and the third to last column will show estimated relative importance of each process in shaping each turnover between communities (samples). The name(s) of the element(s) shows the metrics and its randomization scale, e.g. bNRIiRCa means phylogenetic null model analysis using betaNRI (i.e. SES based on betaMPD) with randomization within each bin and taxonomic null model analysis using RC based on Bray-Curtis with randomization across bins. Other possible phylogenetic null-model-based metrics: bNTI, betaNTI (i.e. SES based on betaMNTD); RCbMPD, RC based on betaMPD; RCbMNTD, RC based on betaMNTD; CbMPD, confidence level based on betaMPD; CbMNTD, confidence level based on betaMNTD. Other possible taxonomic null-model-based metrics: SESbray, SES based on Bray-Curtis; CBray, confidence level based on Bray-Curtis. i, within-bin randomization; a, across-bin randomization.

detail an element in output only if detail.save is TRUE. A list with elements as below.

taxabin an element in 'detail'. A list, show phylogenetic binning results.

The first element is a matrix named sp.bin, where each row is a taxon (OTU or ASV), the first column is the original strict bin ID, the second column is the original bin ID after small bins are merged into nearest relative(s), the third column is the final renewed bin ID.

The second element named bin.united.sp is a list, where each element shows taxa IDs within each bin and the bins are in the order of the final renewed bin IDs.

The third element named `bin.strict.sp` is a list, where each element shows taxa IDs within each strict bin and the bins are in the order of the original strict bin IDs.

The fourth element named `state.strict` is a matrix, where the 1st column is original strict bin IDs, the 2nd column is the taxa number in each strict bin, the 3rd to 5th columns show the maximum, mean, and standard deviation of phylogenetic distances within each strict bin.

The fifth element named `state.united` is a matrix, where the row numbering is the final bin ID, the 1st column is original bin IDs, the 2nd column is the taxa number in each final bin, the 3rd to 5th columns show the maximum, mean, and standard deviation of phylogenetic distances within each final bin.

<code>SigbMPDi</code> , <code>SigbMPDa</code> , <code>SigbMNTDi</code> , <code>SigbMNTDa</code> , <code>SigBCi</code> , <code>SigBCa</code>	elements in 'detail', matrixes showing null model significance testing index for each turnover of each bin. In the name of the element(s), <code>SigbMPD</code> , <code>SigMNTD</code> , or <code>SigBC</code> mean the significance testing is based on <code>betaMPD</code> , <code>betaMNTD</code> , or <code>Bray-Curtis</code> ; <code>i</code> , within-bin randomization; <code>a</code> , across-bin randomization. In each matrix, the first two columns are sample IDs for each turnover; the 3rd to the last column represent different bins with column names containing the significance testing index name, which can be <code>bNRI</code> , <code>bNTI</code> , <code>RCbMPD</code> , <code>RCbMNTD</code> , <code>CbMPD</code> , <code>CbMNTD</code> , <code>SESbray</code> , <code>RCbray</code> , or <code>CBray</code> as mentioned above.
<code>bin.weight</code>	an element in 'detail', a matrix showing relative abundance of each bin in each pair of samples.
<code>processes</code>	an element in 'detail', a list of process importance results at community level.
<code>setting</code>	an element in 'detail', a data.frame showing all basic settings of this function.
<code>comm</code>	an element in 'detail', the input community matrix.
<code>rand</code>	an element in output only if <code>detail.null</code> is <code>TRUE</code> . It is a list with each element showing the observed or null values of a beta diversity index (e.g. <code>betaMPD</code> , <code>betaMNTD</code> , <code>Bray-Curtis</code>). Each index is showed as a list where each element represents a bin.
<code>special.crct</code>	an element in output only if <code>detail.null</code> is <code>TRUE</code> . It shows the corrected values for special cases, where zero means no correction is needed.

Note

Version 8: 2020.10.15, input `comm` as `data.frame` may return error, now include `as.matrix` to solve it. Version 7: 2020.9.21, fix minor bug when `output.wd` is `NULL`. Version 6: 2020.9.1, remove `setwd`; add options to specify some file names; change `dontrun` to `donttest` and revise `save.wd` in help doc. Version 5: 2020.8.19, update help document, add example. Version 4: 2020.5.31. Version 3: 2019.9.30.

Author(s)

Daliang Ning

References

- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Stegen, J.C., Lin, X., Fredrickson, J.K. & Konopka, A.E. (2015). Estimating and mapping ecological processes influencing microbial community assembly. *Front Microbiol*, 6, 370.
- Stegen, J.C., Lin, X., Fredrickson, J.K., Chen, X., Kennedy, D.W., Murray, C.J. et al. (2013). Quantifying community assembly processes and identifying features that impose them. *ISME J*, 7, 2069.
- Zhou, J. & Ning, D. (2017). Stochastic community assembly: Does it matter in microbial ecology? *Microbiology and Molecular Biology Reviews*, 81.
- Veech, J.A. (2012). Significance testing in ecological null models. *Theor Ecol*, 5, 611-616.
- Kane, M.J., Emerson, J., Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[qp.bin.js](#)

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree
# since need to save some output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
pd.wd=paste0(save.wd,"/pdbig")
nworker=4 # parallel computing thread number
rand.time=20 # usually use 1000 for real data.

bin.size.limit=5 # for real data, usually use a proper number
# according to phylogenetic signal test or try some settings
# then choose the reasonable stochasticity level.
# our experience is 12, or 24, or 48.
# but for this example dataset which is too small, have to use 5.

setwd(save.wd)
icamp.out=icamp.big(comm=comm,tree=tree,pd.wd=pd.wd,
                    rand=rand.time, nworker=nworker,
                    bin.size.limit=bin.size.limit)
```

 icamp.bins

 Summarize iCAMP result in each bin

Description

This function is to calculate various statistic index to assess relative importance of each process in each bin and each turnover, and bin's contribution to each process.

Usage

```
icamp.bins(icamp.detail, treat = NULL, clas = NULL, silent = FALSE,
           boot = FALSE, rand.time = 1000, between.group = FALSE)
```

Arguments

icamp.detail	list object, the output or the "detail" element of the output from icamp.big
treat	matrix or data.frame, indicating the group or treatment of each sample, row-names are sample IDs. Allow to input multi-column matrix, different columns represent different ways to group the samples.
clas	matrix or data.frame, the classification information of species (OTUs).
silent	Logic, whether to show messages. Default is FALSE, thus all messages will be showed.
boot	Logic, whether to do bootstrapping test to get significance of dominating process in each bin.
rand.time	integer, bootstrapping times.
between.group	Logic, whether to analyze between-treatment turnovers.

Details

Bin level analysis can provide insights into community assembly mechanisms. This function provides more detailed statistics with the output of the main function [icamp.big](#).

Value

Output is a list object.

Wtuvk	The dominant process in each turnover of each bin.
Ptuv	Relative importance of each process in governing the turnovers between each pair of communities (samples).
Ptk	Relative importance of each process in governing the turnovers of each bin among a group of samples.
Pt	Relative importance of each process in governing the turnovers in a group of samples.
BPtk	Bin contribution to each process, measuring the contribution of each bin to the relative importance of each process in the assembly of a group of communities.

BRPtk	Bin relative contribution to each process, measuring the relative contribution of each bin to a certain process.
Binwt	Output if treat is given. Bin relative abundance in each group (treatment) of samples.
Bin.TopClass	Output if clas is given. A matrix showing the bin relative abundance; the top taxon ID, percentage in bin, and classification; the most abundant name at each phylogeny level in the bin.
Class.Bin	Output if clas is given. A matrix showing the bin ID and classification information for each taxon.

Note

Version 3: 2021.1.5, fix the error when a taxonomy name has unrecognizable character. Version 2: 2020.8.19, update help document, add example. Version 1: 2019.12.11

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[icamp.big](#)

Examples

```
data("icamp.out")
data("example.data")
treatment=example.data$treat
classification=example.data$classification
rand.time=20 # usually use 1000 for real data.
icampbin=icamp.bins(icamp.detail = icamp.out, treat = treatment,
                    clas = classification, boot = TRUE,
                    rand.time = rand.time, between.group = TRUE)
```

icamp.boot

Bootstrapping analysis of icamp results

Description

Use bootstrapping to estimate the variation of relative importance of each process in each group, and compare the difference between groups.

Usage

```
icamp.boot(icamp.result, treat, rand.time = 1000, compare = TRUE,
           silent = FALSE, between.group = FALSE, ST.estimation = FALSE)
```

Arguments

icamp.result	data.frame object, from the output of <code>icamp.big</code> . the first two columns are sample IDs, the third to seventh columns are the relative importance of the five ecological processes.
treat	matrix or data.frame, a one-column (n x 1) matrix indicating the group or treatment of each sample, rownames are sample IDs. if input a n x m matrix, only the first column is used.
rand.time	integer, bootstrapping times. default is 1000.
compare	logic, whether to compare icamp results between different groups.
silent	logic, if FALSE, some messages will show during calculation.
between.group	logic, whether to analyze between-treatment turnovers.
ST.estimation	logic, whether to estimate stochasticity as the total relative importance of dispersal and drift.

Details

Bootstrapping is implemented by random draw samples with replacement, to estimate the variation of relative importance of each process in each group, and calculate the relative difference, effect size, and significance of the difference between each two groups.

Value

Output is a list with three elements.

summary	data.frame, summary of each group. Group: group name from the input "treat". Process: process name from the icamp.result. Observed: the mean relative importance of each process in each group. Mean, Stdev, Min, Quartile25, Median, Quartile75, and Max: mean, standard deviation, minimum, 25 percent-quantile, median, 75 percent-quantile, and maximum of bootstrapping results, respectively. Lower.whisker, Lower.hinge, Mediean.1, Higher.hinge, Higher.whisker, Outlier1...: boxplot elements.
compare	data.frame, summary of comaprison between each two groups. First two columns are group names. From the third column, different indexes for comparison are showed, including Cohen's d (Cohen.d), effect size magnitude according to Cohen's d (Effect.Size), and P value from bootstrapping test (P.value).
boot.detail	a list of matrixes, each matrix corresponds to a group, showing detailed bootstrapping results in each random draw.

Note

Version 3: 2021.1.5, fix error when there is no outlier. Version 2: 2020.8.19, update help document, add example. Version 1: 2019.11.14

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[icamp.big](#)

Examples

```
data("icamp.out")
data("example.data")
treatment=example.data$treat
rand.time=20 # usually use 1000 for real data.
icampbt=icamp.boot(icamp.result = icamp.out$bNRIiRca, treat = treatment,
                  rand.time = rand.time, compare = TRUE,
                  between.group = TRUE, ST.estimation = TRUE)
```

icamp.cate

Summarize iCAMP result for different categories of taxa

Description

This function is to calculate various statistic index to assess relative importance of each process on different categories of taxa. The categories can be defined in various ways. For example, core, consistently and occasionally rare taxa; or different phyla; or various particular functional groups.

Usage

```
icamp.cate(icamp.bins.result, comm, cate, treat = NULL,
          silent = FALSE, between.group = FALSE)
```

Arguments

<code>icamp.bins.result</code>	list object, the output from icamp.bins
<code>comm</code>	matrix or data.frame, community data, each row is a sample or site, each col-name is a taxon (a species or OTU or ASV), thus rownames should be sample IDs, colnames should be taxa IDs.
<code>cate</code>	matrix or data.frame, indicating the category of each taxon, rownames are taxa IDs. If the matrix has multiple columns, only the first column will be used.
<code>treat</code>	matrix or data.frame, indicating the group or treatment of each sample, rownames are sample IDs. Allow to input multi-column matrix, different columns represent different ways to group the samples.
<code>silent</code>	logic, if FALSE, some messages will show during calculation.
<code>between.group</code>	logic, whether to analyze between-treatment turnovers.

Details

This function simply sums up the relative abundance of taxa of a category in different bins governed by a process to summarize the relative importance of the process on the category.

Value

Output is a list object.

<code>Ptuvx</code>	Relative importance of each process in governing each category's turnover between each pair of communities (samples).
<code>Ptx</code>	Relative importance of each process in governing each category's turnovers among a group of samples.

Note

Version 2: 2021.1.7, add help document; fixed NAN error. Version 1: 2020.12.9.

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[icamp.bins](#)

Examples

```

data("icamp.out")
data("example.data")
comm=example.data$comm
treatment=example.data$treat
classification=example.data$classification
rand.time=20 # usually use 1000 for real data.
# 1 # summarize each bin
icampbin=icamp.bins(icamp.detail = icamp.out, treat = treatment,
                   clas = classification, boot = TRUE,
                   rand.time = rand.time, between.group = TRUE)

# 2 # define category
cate=data.frame(type=rep("others",ncol(comm)),stringsAsFactors = FALSE)
rownames(cate)=colnames(comm)
tax.frequency=colSums(comm>0)/nrow(comm)
tax.relative.ab=colMeans(comm/rowSums(comm))
cate[which(tax.frequency>0.75 & tax.relative.ab>0.05),1]="core"
cate[which(tax.relative.ab<0.02),1]="rare"

# 3 # summarize each category
icampcate=icamp.cate(icamp.bins.result = icampbin, comm = comm, cate = cate,
                    treat = treatment, silent = FALSE, between.group = TRUE)

```

icamp.out

*Example output of function icamp.big***Description**

a typical output of icamp.big.

Usage

```
data("icamp.out")
```

Format

The format is: List of 4 \$ bNRIiRCa : 'data.frame': 190 obs. of 7 variables: ..\$ sample1 : Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2\$ sample2 : Factor w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 1\$ Heterogeneous.Selection: num [1:190] 0 0 0 0 0 0 0 0 0 0\$ Homogeneous.Selection : num [1:190] 0 0 0 0 0\$ Dispersal.Limitation : num [1:190] 0 0 0 0 0 0 0 0 0\$ Homogenizing.Dispersal : num [1:190] 0 0.649 0.628 0 0\$ Drift.and.Others : num [1:190] 1 0.351 0.372 1 1 ... \$ detail :List of 7 ..\$ taxabin :List of 5 ..\$ sp.bin : 'data.frame': 30 obs. of 3 variables:\$ bin.id.strict: int [1:30] 1 2 3 4 5 6 7 8 9 10\$ bin.id.united: chr [1:30] "34" "34" "34" "34"\$ bin.id.new : num [1:30] 1 1 1 1 1 2 2 2 2 2\$ bin.united.sp:List of 3\$: chr [1:5] "OTU1" "OTU2" "OTU3" "OTU4"\$: chr [1:7] "OTU6" "OTU7" "OTU8" "OTU9"\$: chr [1:18] "OTU13" "OTU14" "OTU15" "OTU16"\$ bin.strict.sp:List of 30\$ 1 : chr "OTU1"\$ 2 : chr "OTU2"\$ 3 :


```
0.0949 0.047 ... .. attr(*, "dimnames")=List of 2 ... ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..
... ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..$ : num [1:20, 1:20] 0 1.48 1.29 1.27 1.64 ... ..$ : chr [1:20] "s1" "s2"
"dimnames")=List of 2 ... ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..$ : chr [1:20] "s1" "s2"
"s3" "s4" ... ..$ bMPPDi.rand:List of 3 ... ..$ : 'data.frame': 190 obs. of 102 variables: ... ..$ name1 :
Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... ..$ name2 : Factor w/
19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... ..$ rand1 : num [1:190] 0.00798 0.03197
0.01217 0.01476 0.00992 ... ..$ rand2 : num [1:190] 0.00543 0.02743 0.01838 0.01224 0.00822
... ..$ rand3 : num [1:190] 0.00357 0.0245 0.02358 0.01224 0.00822 ... ..$ rand4 : num
[1:190] 0.00839 0.04672 0.03604 0.00615 0.00413 ... ..$ rand5 : num [1:190] 0.0016 0.0226
0.0313 0.0172 0.0116 ... ..$ rand6 : num [1:190] 0.00626 0.03441 0.0261 0.00615 0.00413 ... ..
..$ rand7 : num [1:190] 0.00839 0.03746 0.01959 0.01476 0.00992 ... ..$ rand8 : num [1:190]
0.00839 0.03351 0.01257 0.01722 0.01157 ... ..$ rand9 : num [1:190] 0.00798 0.03268 0.01343
0.00697 0.00468 ... ..$ rand10 : num [1:190] 0.00357 0.02626 0.02671 0.00615 0.00413 ... ..
..$ rand11 : num [1:190] 0.00357 0.03161 0.03623 0.00313 0.0021 ... ..$ rand12 : num [1:190]
0.0016 0.0125 0.01325 0.01476 0.00992 ... ..$ rand13 : num [1:190] 0.00881 0.05001 0.03956
0.00248 0.00166 ... ..$ rand14 : num [1:190] 0.00127 0.02228 0.03252 0.01722 0.01157 ...
.. ..$ rand15 : num [1:190] 0.00881 0.04554 0.03161 0.00248 0.00166 ... ..$ rand16 :
num [1:190] 0.0016 0.02264 0.03127 0.00697 0.00468 ... ..$ rand17 : num [1:190] 0.00881
0.03988 0.02154 0.00313 0.0021 ... ..$ rand18 : num [1:190] 0.0016 0.02176 0.02971 0.00615
0.00413 ... ..$ rand19 : num [1:190] 0.0016 0.0134 0.0148 0.0172 0.0116 ... ..$ rand20 :
num [1:190] 0.00127 0.02053 0.02939 0.01558 0.01047 ... ..$ rand21 : num [1:190] 0.00543
0.03581 0.03327 0.00248 0.00166 ... ..$ rand22 : num [1:190] 0.00798 0.04518 0.03565 0.0106
0.00712 ... ..$ rand23 : num [1:190] 0.00798 0.03197 0.01217 0.0106 0.00712 ... ..$ rand24
: num [1:190] 0.00881 0.05001 0.03956 0.01224 0.00822 ... ..$ rand25 : num [1:190] 0.00543
0.03309 0.02845 0.00697 0.00468 ... ..$ rand26 : num [1:190] 0.00839 0.03421 0.01382
0.00615 0.00413 ... ..$ rand27 : num [1:190] 0.00357 0.02626 0.02671 0.0106 0.00712 ... ..
..$ rand28 : num [1:190] 0.00881 0.03575 0.01421 0.00697 0.00468 ... ..$ rand29 : num
[1:190] 0.00626 0.03441 0.0261 0.00697 0.00468 ... ..$ rand30 : num [1:190] 0.00798 0.04782
0.04035 0.00697 0.00468 ... ..$ rand31 : num [1:190] 0.00626 0.04064 0.03718 0.00248
0.00166 ... ..$ rand32 : num [1:190] 0.00127 0.02316 0.03408 0.01224 0.00822 ... ..$
rand33 : num [1:190] 0.00626 0.04152 0.03875 0.00248 0.00166 ... ..$ rand34 : num [1:190]
0.00881 0.03575 0.01421 0.01558 0.01047 ... ..$ rand35 : num [1:190] 0.0016 0.0244 0.0344
0.0156 0.0105 ... ..$ rand36 : num [1:190] 0.00543 0.02655 0.01681 0.01224 0.00822 ... ..
..$ rand37 : num [1:190] 0.00127 0.01781 0.02457 0.0164 0.01102 ... ..$ rand38 : num [1:190]
0.00756 0.03916 0.02731 0.00248 0.00166 ... ..$ rand39 : num [1:190] 0.00756 0.03438 0.0188
0.00313 0.0021 ... ..$ rand40 : num [1:190] 0.00357 0.01647 0.00931 0.01722 0.01157 ... ..
..$ rand41 : num [1:190] 0.00127 0.02053 0.02939 0.0106 0.00712 ... ..$ rand42 : num [1:190]
0.0016 0.0125 0.0132 0.0164 0.011 ... ..$ rand43 : num [1:190] 0.0016 0.0218 0.0297 0.0164
0.011 ... ..$ rand44 : num [1:190] 0.00756 0.0454 0.03839 0.00313 0.0021 ... ..$ rand45
: num [1:190] 0.0016 0.02352 0.03283 0.00615 0.00413 ... ..$ rand46 : num [1:190] 0.00626
0.02963 0.0176 0.00697 0.00468 ... ..$ rand47 : num [1:190] 0.00315 0.02472 0.02632 0.0106
0.00712 ... ..$ rand48 : num [1:190] 0.00798 0.04782 0.04035 0.00313 0.0021 ... ..$ rand49
: num [1:190] 0.00315 0.02744 0.03115 0.0164 0.01102 ... ..$ rand50 : num [1:190] 0.00839
0.03746 0.01959 0.00313 0.0021 ... ..$ rand51 : num [1:190] 0.00127 0.02141 0.03096 0.0106
0.00712 ... ..$ rand52 : num [1:190] 0.00839 0.03421 0.01382 0.01476 0.00992 ... ..$
rand53 : num [1:190] 0.00839 0.04935 0.04074 0.01224 0.00822 ... ..$ rand54 : num [1:190]
0.00315 0.02919 0.03427 0.00313 0.0021 ... ..$ rand55 : num [1:190] 0.00357 0.02986 0.0331
0.00313 0.0021 ... ..$ rand56 : num [1:190] 0.00357 0.01647 0.00931 0.01558 0.01047 ... ..
```

```
.. ..$ rand57 : num [1:190] 0.00315 0.02296 0.02319 0.00697 0.00468 ... ..$ rand58 : num
[1:190] 0.00756 0.0454 0.03839 0.00615 0.00413 ... ..$ rand59 : num [1:190] 0.00315 0.01493
0.00892 0.0164 0.01102 ... ..$ rand60 : num [1:190] 0.00315 0.02744 0.03115 0.00313 0.0021
... ..$ rand61 : num [1:190] 0.00756 0.03438 0.0188 0.0164 0.01102 ... ..$ rand62 : num
[1:190] 0.00839 0.044 0.03122 0.00248 0.00166 ... ..$ rand63 : num [1:190] 0.00798 0.0368
0.02076 0.00313 0.0021 ... ..$ rand64 : num [1:190] 0.00315 0.01493 0.00892 0.01476 0.00992
... ..$ rand65 : num [1:190] 0.00798 0.04518 0.03565 0.00248 0.00166 ... ..$ rand66 : num
[1:190] 0.00626 0.02963 0.0176 0.0106 0.00712 ... ..$ rand67 : num [1:190] 0.00881 0.03505
0.01296 0.0164 0.01102 ... ..$ rand68 : num [1:190] 0.00626 0.04152 0.03875 0.0164 0.01102
... ..$ rand69 : num [1:190] 0.00626 0.02568 0.01058 0.0164 0.01102 ... ..$ rand70 : num
[1:190] 0.00127 0.02141 0.03096 0.01476 0.00992 ... ..$ rand71 : num [1:190] 0.00798 0.0407
0.0277 0.01476 0.00992 ... ..$ rand72 : num [1:190] 0.00839 0.04935 0.04074 0.00248 0.00166
... ..$ rand73 : num [1:190] 0.00543 0.02743 0.01838 0.00615 0.00413 ... ..$ rand74 :
num [1:190] 0.00881 0.03505 0.01296 0.01224 0.00822 ... ..$ rand75 : num [1:190] 0.00626
0.04064 0.03718 0.01722 0.01157 ... ..$ rand76 : num [1:190] 0.00881 0.03988 0.02154
0.01558 0.01047 ... ..$ rand77 : num [1:190] 0.00315 0.01906 0.01625 0.01224 0.00822 ... ..
..$ rand78 : num [1:190] 0.00756 0.03114 0.01304 0.00615 0.00413 ... ..$ rand79 : num [1:190]
0.0016 0.02352 0.03283 0.01476 0.00992 ... ..$ rand80 : num [1:190] 0.00127 0.01605 0.02144
0.01476 0.00992 ... ..$ rand81 : num [1:190] 0.00127 0.01605 0.02144 0.01558 0.01047 ... ..
..$ rand82 : num [1:190] 0.00543 0.03669 0.03484 0.01476 0.00992 ... ..$ rand83 : num
[1:190] 0.00626 0.03051 0.01916 0.0106 0.00712 ... ..$ rand84 : num [1:190] 0.00798 0.0368
0.02076 0.01722 0.01157 ... ..$ rand85 : num [1:190] 0.00881 0.04914 0.038 0.00313 0.0021
... ..$ rand86 : num [1:190] 0.00357 0.02986 0.0331 0.01722 0.01157 ... ..$ rand87 : num
[1:190] 0.00756 0.03043 0.01178 0.0106 0.00712 ... ..$ rand88 : num [1:190] 0.00756 0.03114
0.01304 0.0164 0.01102 ... ..$ rand89 : num [1:190] 0.00543 0.0226 0.00979 0.01558 0.01047
... ..$ rand90 : num [1:190] 0.00626 0.02568 0.01058 0.01722 0.01157 ... ..$ rand91 : num
[1:190] 0.00127 0.02316 0.03408 0.0164 0.01102 ... ..$ rand92 : num [1:190] 0.00756 0.03916
0.02731 0.01558 0.01047 ... ..$ rand93 : num [1:190] 0.00881 0.04914 0.038 0.00697 0.00468
... ..$ rand94 : num [1:190] 0.00756 0.04452 0.03683 0.00248 0.00166 ... ..$ rand95 :
num [1:190] 0.0016 0.0134 0.0148 0.0156 0.0105 ... ..$ rand96 : num [1:190] 0.00357 0.01972
0.01508 0.0106 0.00712 ... ..$ rand97 : num [1:190] 0.00543 0.03309 0.02845 0.00615 0.00413
... ..$ [list output truncated] ..$ : 'data.frame': 190 obs. of 102 variables: ..$ name1 : Factor
w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... ..$ name2 : Factor w/ 19 levels
"s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... ..$ rand1 : num [1:190] 0.1008 0.0194 0.1046 0.0538
0 ... ..$ rand2 : num [1:190] 0.1009 0.0418 0.0792 0.059 0 ... ..$ rand3 : num [1:190]
0.093 0.0289 0.0847 0.0521 0 ... ..$ rand4 : num [1:190] 0.0695 0.0198 0.0566 0.0386 0 ... ..
..$ rand5 : num [1:190] 0.1207 0.0288 0.1134 0.0656 0 ... ..$ rand6 : num [1:190] 0.0985
0.0328 0.0897 0.0557 0 ... ..$ rand7 : num [1:190] 0.0769 0.0274 0.0749 0.0439 0 ... ..$
rand8 : num [1:190] 0.0659 0.0216 0.0713 0.0372 0 ... ..$ rand9 : num [1:190] 0.1118 0.0418
0.1078 0.0643 0 ... ..$ rand10 : num [1:190] 0.1054 0.0256 0.0966 0.0574 0 ... ..$ rand11 :
num [1:190] 0.0823 0.0288 0.0721 0.0469 0 ... ..$ rand12 : num [1:190] 0.1124 0.0274 0.0929
0.0613 0 ... ..$ rand13 : num [1:190] 0.1193 0.044 0.1063 0.0685 0 ... ..$ rand14 : num
[1:190] 0.1036 0.0384 0.094 0.0595 0 ... ..$ rand15 : num [1:190] 0.0938 0.0271 0.0728 0.0521
0 ... ..$ rand16 : num [1:190] 0.1036 0.0384 0.0924 0.0595 0 ... ..$ rand17 : num [1:190]
0.1112 0.0385 0.0964 0.0633 0 ... ..$ rand18 : num [1:190] 0.0881 0.0328 0.0837 0.0506 0 ... ..
..$ rand19 : num [1:190] 0.0876 0.0271 0.0684 0.0491 0 ... ..$ rand20 : num [1:190] 0.0846
0.0198 0.0862 0.0459 0 ... ..$ rand21 : num [1:190] 0.1051 0.0148 0.101 0.0548 0 ... ..$
rand22 : num [1:190] 0.1387 0.0384 0.1101 0.0766 0 ... ..$ rand23 : num [1:190] 0.08 0.0295
```


0.0768 0.0459 0\$ rand24 : num [1:190] 0.1174 0.0311 0.1051 0.0646 0\$ rand25 :
num [1:190] 0.0883 0.0256 0.0691 0.0491 0\$ rand26 : num [1:190] 0.1071 0.0385 0.088
0.0613 0\$ rand27 : num [1:190] 0.1282 0.0311 0.1127 0.0699 0\$ rand28 : num
[1:190] 0.1254 0.044 0.12 0.0715 0\$ rand29 : num [1:190] 0.1009 0.0418 0.0917 0.059
0\$ rand30 : num [1:190] 0.1054 0.0194 0.0979 0.056 0\$ rand31 : num [1:190]
0.099 0.0328 0.0919 0.056 0\$ rand32 : num [1:190] 0.0938 0.0271 0.0728 0.0521 0
..\$ rand33 : num [1:190] 0.1008 0.0311 0.1012 0.0565 0\$ rand34 : num [1:190] 0.1147
0.0418 0.0949 0.0657 0\$ rand35 : num [1:190] 0.0825 0.0289 0.0677 0.047 0\$
rand36 : num [1:190] 0.1031 0.0328 0.0926 0.058 0\$ rand37 : num [1:190] 0.0899 0.0385
0.0949 0.0529 0\$ rand38 : num [1:190] 0.1104 0.0216 0.1006 0.059 0\$ rand39 :
num [1:190] 0.0869 0.0194 0.0928 0.047 0\$ rand40 : num [1:190] 0.0501 0.0198 0.0605
0.0291 0\$ rand41 : num [1:190] 0.094 0.0296 0.0917 0.0528 0\$ rand42 : num
[1:190] 0.0865 0.0407 0.0823 0.0517 0\$ rand43 : num [1:190] 0.1221 0.044 0.1069 0.0699
0\$ rand44 : num [1:190] 0.0541 0.0114 0.0642 0.0291 0\$ rand45 : num [1:190]
0.0805 0.0198 0.0782 0.0439 0\$ rand46 : num [1:190] 0.0934 0.0216 0.1001 0.0506 0 ...
..\$ rand47 : num [1:190] 0.094 0.0274 0.0969 0.0522 0\$ rand48 : num [1:190] 0.1015
0.0296 0.085 0.0565 0\$ rand49 : num [1:190] 0.1015 0.0296 0.1019 0.0565 0\$
rand50 : num [1:190] 0.0908 0.0198 0.0857 0.049 0\$ rand51 : num [1:190] 0.0879 0.0418
0.0761 0.0527 0\$ rand52 : num [1:190] 0.1118 0.0271 0.0854 0.0609 0\$ rand53 :
num [1:190] 0.1032 0.0361 0.0784 0.0588 0\$ rand54 : num [1:190] 0.0794 0.0296 0.0885
0.0456 0\$ rand55 : num [1:190] 0.1112 0.0384 0.0909 0.0632 0\$ rand56 : num
[1:190] 0.133 0.0418 0.114 0.0746 0\$ rand57 : num [1:190] 0.0988 0.0148 0.0938 0.0517
0\$ rand58 : num [1:190] 0.1199 0.0296 0.1148 0.0655 0\$ rand59 : num [1:190]
0.1435 0.0418 0.1215 0.0798 0\$ rand60 : num [1:190] 0.1249 0.0361 0.1007 0.0694 0
... ..\$ rand61 : num [1:190] 0.0917 0.0256 0.0855 0.0508 0\$ rand62 : num [1:190]
0.1032 0.0361 0.094 0.0588 0\$ rand63 : num [1:190] 0.1252 0.0289 0.1136 0.0679 0
..\$ rand64 : num [1:190] 0.0965 0.0194 0.1025 0.0517 0\$ rand65 : num [1:190] 0.0766
0.0271 0.0768 0.0437 0\$ rand66 : num [1:190] 0.0995 0.0184 0.1038 0.0529 0
..\$ rand67 : num [1:190] 0.1209 0.0289 0.1205 0.0657 0\$ rand68 : num [1:190] 0.1118
0.0418 0.1078 0.0643 0\$ rand69 : num [1:190] 0.068 0.0114 0.0691 0.0358 0\$
rand70 : num [1:190] 0.0674 0.0184 0.0708 0.0372 0\$ rand71 : num [1:190] 0.1124 0.0274
0.1128 0.0613 0\$ rand72 : num [1:190] 0.0944 0.0296 0.092 0.053 0\$ rand73 :
num [1:190] 0.1237 0.0385 0.1194 0.0694 0\$ rand74 : num [1:190] 0.0801 0.0296 0.0771
0.046 0\$ rand75 : num [1:190] 0.0944 0.0296 0.0805 0.053 0\$ rand76 : num
[1:190] 0.1032 0.0361 0.1024 0.0588 0\$ rand77 : num [1:190] 0.1009 0.0418 0.0917 0.059
0\$ rand78 : num [1:190] 0.0733 0.0361 0.0701 0.0442 0\$ rand79 : num [1:190]
0.1051 0.0148 0.0911 0.0548 0\$ rand80 : num [1:190] 0.0822 0.0311 0.0797 0.0474 0 ...
..\$ rand81 : num [1:190] 0.0986 0.0295 0.0834 0.055 0\$ rand82 : num [1:190] 0.0865
0.0407 0.0823 0.0517 0\$ rand83 : num [1:190] 0.0659 0.0216 0.0713 0.0372 0\$
rand84 : num [1:190] 0.1038 0.0385 0.0946 0.0596 0\$ rand85 : num [1:190] 0.0999 0.044
0.0991 0.059 0\$ rand86 : num [1:190] 0.1174 0.0407 0.0949 0.0668 0\$ rand87 :
num [1:190] 0.1326 0.0385 0.1113 0.0737 0\$ rand88 : num [1:190] 0.0846 0.0198 0.0811
0.0459 0\$ rand89 : num [1:190] 0.0769 0.0274 0.0749 0.0439 0\$ rand90 : num
[1:190] 0.0755 0.0114 0.0772 0.0395 0\$ rand91 : num [1:190] 0.136 0.044 0.1166 0.0766
0\$ rand92 : num [1:190] 0.1036 0.0384 0.0924 0.0595 0\$ rand93 : num [1:190]
0.0501 0.0198 0.0491 0.0291 0\$ rand94 : num [1:190] 0.0562 0.0148 0.0739 0.0309 0
... ..\$ rand95 : num [1:190] 0.0902 0.0289 0.084 0.0508 0\$ rand96 : num [1:190]
0.086 0.0216 0.0869 0.047 0\$ rand97 : num [1:190] 0.093 0.0289 0.0847 0.0521 0

```

.. [list output truncated] .. .$ : 'data.frame': 190 obs. of 102 variables: .. ..$ name1 : Factor w/
19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... .. ..$ name2 : Factor w/ 19 levels
"s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... .. ..$ rand1 : num [1:190] 1.36 1.25 1.23 1.66 2.19 ... ..
.. ..$ rand2 : num [1:190] 1.46 1.25 1.27 1.79 2.31 ... .. ..$ rand3 : num [1:190] 1.34 1.15 1.06
1.52 1.96 ... .. ..$ rand4 : num [1:190] 1.26 1.13 1.03 1.43 1.94 ... .. ..$ rand5 : num [1:190]
1.349 1.137 0.997 1.545 1.841 ... .. ..$ rand6 : num [1:190] 1.43 1.27 1.23 1.67 2.1 ... .. ..$
rand7 : num [1:190] 1.12 1.02 1.04 1.48 2.07 ... .. ..$ rand8 : num [1:190] 1.38 1.17 1.17 1.8 2.34
... .. ..$ rand9 : num [1:190] 1.15 0.934 0.867 1.5 1.997 ... .. ..$ rand10 : num [1:190] 1.48 1.26
1.22 1.75 2.26 ... .. ..$ rand11 : num [1:190] 1.39 1.25 1.22 1.48 1.98 ... .. ..$ rand12 : num
[1:190] 1.28 1.11 1.17 1.87 2.41 ... .. ..$ rand13 : num [1:190] 1.328 1.109 0.973 1.653 2.036 ...
.. ..$ rand14 : num [1:190] 1.5 1.3 1.28 1.62 2.06 ... .. ..$ rand15 : num [1:190] 1.35 1.24 1.27
1.74 2.27 ... .. ..$ rand16 : num [1:190] 0.99 0.865 0.923 1.242 1.662 ... .. ..$ rand17 : num
[1:190] 1.26 1.15 1.08 1.52 1.98 ... .. ..$ rand18 : num [1:190] 1.259 1.08 0.976 1.326 1.75 ... ..
.. ..$ rand19 : num [1:190] 1.72 1.53 1.44 2.03 2.62 ... .. ..$ rand20 : num [1:190] 1.085 0.868
0.893 1.23 1.578 ... .. ..$ rand21 : num [1:190] 0.969 0.938 0.883 1.278 1.695 ... .. ..$ rand22
: num [1:190] 1.17 1.03 1 1.4 1.81 ... .. ..$ rand23 : num [1:190] 1.26 1.04 1.06 1.36 1.72 ... ..
.. ..$ rand24 : num [1:190] 1.144 0.993 0.991 1.479 1.978 ... .. ..$ rand25 : num [1:190] 0.973
0.818 0.878 1.294 1.771 ... .. ..$ rand26 : num [1:190] 1.44 1.29 1.25 1.82 2.33 ... .. ..$ rand27
: num [1:190] 1.34 1.23 1.22 1.55 1.9 ... .. ..$ rand28 : num [1:190] 1.21 1.12 1.11 1.6 2.06 ...
.. ..$ rand29 : num [1:190] 0.996 0.869 0.977 1.239 1.724 ... .. ..$ rand30 : num [1:190] 1.29
1.04 1.03 1.59 2 ... .. ..$ rand31 : num [1:190] 0.991 0.807 0.907 1.473 1.771 ... .. ..$ rand32
: num [1:190] 1.59 1.3 1.27 1.7 2.23 ... .. ..$ rand33 : num [1:190] 1.52 1.44 1.3 1.92 2.46 ... ..
.. ..$ rand34 : num [1:190] 1.12 1.01 1 1.41 1.77 ... .. ..$ rand35 : num [1:190] 1.76 1.55 1.51
1.88 2.43 ... .. ..$ rand36 : num [1:190] 1.13 1.02 1.08 1.43 1.92 ... .. ..$ rand37 : num [1:190]
1.57 1.25 1.39 1.91 2.4 ... .. ..$ rand38 : num [1:190] 1.4 1.24 1.24 1.66 2.08 ... .. ..$ rand39 :
num [1:190] 1.43 1.18 1.13 1.66 2.15 ... .. ..$ rand40 : num [1:190] 1.78 1.5 1.38 1.84 2.31 ... ..
.. ..$ rand41 : num [1:190] 1.33 1.09 1.06 1.55 1.94 ... .. ..$ rand42 : num [1:190] 1.45 1.33 1.22
1.76 2.31 ... .. ..$ rand43 : num [1:190] 1.119 1.014 0.993 1.496 1.879 ... .. ..$ rand44 : num
[1:190] 1.4 1.29 1.18 1.63 2.17 ... .. ..$ rand45 : num [1:190] 1.59 1.45 1.29 1.68 2.2 ... .. ..$
rand46 : num [1:190] 1.45 1.28 1.27 1.77 2.22 ... .. ..$ rand47 : num [1:190] 1.24 1.05 1.09 1.5
2.03 ... .. ..$ rand48 : num [1:190] 1.14 1.16 1.17 1.54 2.1 ... .. ..$ rand49 : num [1:190] 0.951
0.928 0.924 1.384 1.856 ... .. ..$ rand50 : num [1:190] 1.39 1.24 1.23 1.65 2.05 ... .. ..$ rand51
: num [1:190] 1.233 1.037 0.956 1.522 1.937 ... .. ..$ rand52 : num [1:190] 0.962 0.815 0.929
1.202 1.568 ... .. ..$ rand53 : num [1:190] 1.42 1.3 1.28 1.95 2.46 ... .. ..$ rand54 : num [1:190]
0.936 0.733 0.769 1.24 1.453 ... .. ..$ rand55 : num [1:190] 1.197 0.968 0.983 1.272 1.743 ... ..
.. ..$ rand56 : num [1:190] 1.48 1.37 1.29 1.7 2.33 ... .. ..$ rand57 : num [1:190] 1.038 0.976
1.023 1.476 1.916 ... .. ..$ rand58 : num [1:190] 1.42 1.3 1.2 1.69 2.18 ... .. ..$ rand59 : num
[1:190] 0.979 0.93 0.833 1.418 1.849 ... .. ..$ rand60 : num [1:190] 1.33 1.11 1.15 1.63 2.09 ... ..
.. ..$ rand61 : num [1:190] 1.54 1.23 1.13 1.59 2.07 ... .. ..$ rand62 : num [1:190] 1.117 0.805
0.844 1.329 1.576 ... .. ..$ rand63 : num [1:190] 1.25 1.12 1.21 1.44 1.82 ... .. ..$ rand64 :
num [1:190] 1.36 1.18 1.13 1.6 2.09 ... .. ..$ rand65 : num [1:190] 1.15 1.029 0.947 1.348 1.851
... .. ..$ rand66 : num [1:190] 1.143 0.963 0.956 1.278 1.784 ... .. ..$ rand67 : num [1:190]
1.128 0.834 0.855 1.401 1.661 ... .. ..$ rand68 : num [1:190] 1.02 0.93 0.87 1.57 1.97 ... .. ..$
rand69 : num [1:190] 0.984 0.931 0.919 1.472 1.913 ... .. ..$ rand70 : num [1:190] 1.092 0.835
0.889 1.337 1.635 ... .. ..$ rand71 : num [1:190] 1.2 1.104 0.992 1.777 2.262 ... .. ..$ rand72 :
num [1:190] 1.21 1.06 1.14 1.27 1.64 ... .. ..$ rand73 : num [1:190] 1.32 1.07 0.93 1.55 2.07 ... ..
.. ..$ rand74 : num [1:190] 1.5 1.3 1.26 1.74 2.27 ... .. ..$ rand75 : num [1:190] 1.62 1.46 1.42
1.86 2.45 ... .. ..$ rand76 : num [1:190] 1.28 1.09 1.13 1.7 2.2 ... .. ..$ rand77 : num [1:190]

```

```

1.156 0.874 0.936 1.548 2.069 ... ..$ rand78 : num [1:190] 1.201 0.989 0.922 1.352 1.769 ... ..
..$ rand79 : num [1:190] 1.34 1.12 1.14 1.37 1.8 ... ..$ rand80 : num [1:190] 1.31 1.12 1.02
1.59 2.06 ... ..$ rand81 : num [1:190] 1.43 1.13 1.03 1.49 1.94 ... ..$ rand82 : num [1:190]
1.216 0.798 0.897 1.465 1.809 ... ..$ rand83 : num [1:190] 1.139 1.048 0.957 1.462 1.843 ... ..
..$ rand84 : num [1:190] 1.5 1.23 1.18 1.76 2.26 ... ..$ rand85 : num [1:190] 1.58 1.46 1.45
1.87 2.53 ... ..$ rand86 : num [1:190] 1.39 1.22 1.18 1.65 2.14 ... ..$ rand87 : num [1:190]
0.994 0.731 0.766 1.279 1.735 ... ..$ rand88 : num [1:190] 1.15 1.04 1.06 1.54 2.01 ... ..$
rand89 : num [1:190] 1.19 1.02 1.05 1.46 1.84 ... ..$ rand90 : num [1:190] 1.65 1.39 1.35 1.93
2.29 ... ..$ rand91 : num [1:190] 1.26 1.01 1.04 1.39 1.81 ... ..$ rand92 : num [1:190] 1.57
1.35 1.3 1.76 2.16 ... ..$ rand93 : num [1:190] 1.22 1.09 1.11 1.31 1.74 ... ..$ rand94 : num
[1:190] 1.4 1.31 1.24 1.78 2.35 ... ..$ rand95 : num [1:190] 1.71 1.48 1.36 1.92 2.48 ... ..$
rand96 : num [1:190] 1.4 1.23 1.04 1.61 2.02 ... ..$ rand97 : num [1:190] 1.47 1.23 1.18 1.5
1.96 ... .. [list output truncated] ..$ BC.obs :List of 3 ..$ : num [1:20, 1:20] 0 0.0317 0.1653
0.1389 0.051 ... ..$ attr(*, "dimnames")=List of 2 ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..
..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..$ : num [1:20, 1:20] 0 0.1468 0.0861 0.1444 0.1111
... ..$ attr(*, "dimnames")=List of 2 ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..$ : chr
[1:20] "s1" "s2" "s3" "s4" ... ..$ : num [1:20, 1:20] 0 0.345 0.263 0.317 0.494 ... ..$ attr(*,
"dimnames")=List of 2 ..$ : chr [1:20] "s1" "s2" "s3" "s4" ... ..$ : chr [1:20] "s1" "s2"
"s3" "s4" ... ..$ BCa.rand :List of 3 ..$ : 'data.frame': 190 obs. of 102 variables: ..$ name1
: Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... ..$ name2 : Factor
w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... ..$ rand1 : num [1:190] 0.127 0.1931
0.0333 0.137 0.0451 ... ..$ rand2 : num [1:190] 0.1429 0.0875 0.0778 0.0465 0.0208 ... ..$
rand3 : num [1:190] 0.1071 0.125 0.0667 0.0698 0.0521 ... ..$ rand4 : num [1:190] 0.19 0.225
0.206 0.262 0.281 ... ..$ rand5 : num [1:190] 0.107 0.296 0.117 0.095 0.104 ... ..$ rand6 :
num [1:190] 0.123 0.169 0.194 0.148 0.163 ... ..$ rand7 : num [1:190] 0 0.1125 0.0889 0.0698
0.0104 ... ..$ rand8 : num [1:190] 0.107 0.167 0.178 0.143 0.104 ... ..$ rand9 : num [1:190]
0.127 0.1431 0.0778 0.1253 0.0972 ... ..$ rand10 : num [1:190] 0.1587 0.1403 0.0278 0.1208
0.0903 ... ..$ rand11 : num [1:190] 0.1548 0.1958 0.0944 0.1415 0.0938 ... ..$ rand12 :
num [1:190] 0.0833 0.0375 0.1111 0.093 0.1562 ... ..$ rand13 : num [1:190] 0.111 0.211 0.189
0.123 0.142 ... ..$ rand14 : num [1:190] 0.0992 0.1361 0.1556 0.1693 0.2049 ... ..$ rand15
: num [1:190] 0.127 0.156 0.1 0.16 0.16 ... ..$ rand16 : num [1:190] 0.07937 0.07778 0.00556
0.13243 0.07986 ... ..$ rand17 : num [1:190] 0.234 0.16 0.244 0.222 0.253 ... ..$ rand18
: num [1:190] 0.0754 0.1653 0.15 0.1441 0.0278 ... ..$ rand19 : num [1:190] 0.1905 0.1125
0.1556 0.093 0.0104 ... ..$ rand20 : num [1:190] 0.337 0.29 0.3 0.243 0.236 ... ..$ rand21
: num [1:190] 0.1706 0.1736 0.1889 0.0995 0.1944 ... ..$ rand22 : num [1:190] 0.0952 0.0375
0.1889 0.1395 0.0833 ... ..$ rand23 : num [1:190] 0.0595 0.1333 0.1722 0.1531 0.059 ... ..
..$ rand24 : num [1:190] 0.19 0.204 0.178 0.167 0.115 ... ..$ rand25 : num [1:190] 0.0952 0.05
0.1889 0.093 0.0417 ... ..$ rand26 : num [1:190] 0.246 0.21 0.267 0.257 0.264 ... ..$ rand27
: num [1:190] 0.1786 0.2125 0.0222 0.0698 0.125 ... ..$ rand28 : num [1:190] 0.2262 0.1292
0.0556 0.1085 0.1875 ... ..$ rand29 : num [1:190] 0.119 0.075 0.0778 0.0465 0.1042 ... ..
..$ rand30 : num [1:190] 0.0595 0.075 0.1 0.1163 0.125 ... ..$ rand31 : num [1:190] 0.119 0.1
0.167 0.128 0 ... ..$ rand32 : num [1:190] 0.131 0.0625 0.1667 0.0814 0.0833 ... ..$ rand33
: num [1:190] 0.0119 0.0958 0.1389 0.1996 0.1667 ... ..$ rand34 : num [1:190] 0.119 0 0.1444
0.0465 0.125 ... ..$ rand35 : num [1:190] 0.23 0.249 0.144 0.111 0.132 ... ..$ rand36 : num
[1:190] 0.0357 0.15 0.0333 0.0233 0.1146 ... ..$ rand37 : num [1:190] 0.0675 0.0681 0.0889
0.1486 0.1389 ... ..$ rand38 : num [1:190] 0.0119 0.1 0.1444 0.1279 0.125 ... ..$ rand39
: num [1:190] 0.0873 0.1653 0.1278 0.051 0.1528 ... ..$ rand40 : num [1:190] 0.246 0.222
0.233 0.257 0.243 ... ..$ rand41 : num [1:190] 0.119 0.125 0.0778 0.0814 0.0521 ... ..$

```

```

rand42 : num [1:190] 0.0714 0.125 0.1333 0.0698 0.1562 ... ..$ rand43 : num [1:190] 0.0595
0.1 0.0778 0.0698 0.0729 ... ..$ rand44 : num [1:190] 0.1429 0.15 0.0333 0.0814 0.1042 ... ..
..$ rand45 : num [1:190] 0.0476 0.1 0.1444 0.0814 0.0938 ... ..$ rand46 : num [1:190] 0.1667
0.075 0.0111 0.093 0.0521 ... ..$ rand47 : num [1:190] 0.0317 0.1306 0.1667 0.0904 0.0556 ...
.. ..$ rand48 : num [1:190] 0.119 0.1125 0.0667 0.0349 0.1562 ... ..$ rand49 : num [1:190]
0.0476 0 0.0444 0.0465 0.1354 ... ..$ rand50 : num [1:190] 0.163 0.133 0.161 0.095 0.125 ... ..
..$ rand51 : num [1:190] 0.0595 0.125 0.1 0.0698 0.1042 ... ..$ rand52 : num [1:190] 0.1667
0.15 0.0889 0.1628 0 ... ..$ rand53 : num [1:190] 0.1667 0 0.0889 0.093 0 ... ..$ rand54
: num [1:190] 0.0357 0.125 0.1111 0.1163 0.1354 ... ..$ rand55 : num [1:190] 0.0873 0.1167
0.1667 0.1667 0.1562 ... ..$ rand56 : num [1:190] 0.1587 0.2736 0.1 0.0762 0.1632 ... ..$
rand57 : num [1:190] 0.0556 0.1181 0.0667 0.084 0.0278 ... ..$ rand58 : num [1:190] 0.0833
0.1875 0.1 0.1279 0.0938 ... ..$ rand59 : num [1:190] 0.1111 0.0736 0.1778 0.0879 0.1632 ...
.. ..$ rand60 : num [1:190] 0.1746 0.1639 0.1833 0.0691 0.1424 ... ..$ rand61 : num [1:190]
0 0.1375 0.1222 0.093 0.0833 ... ..$ rand62 : num [1:190] 0.1151 0.1403 0.0833 0.0536 0.0799
... ..$ rand63 : num [1:190] 0.135 0.178 0.106 0.121 0.17 ... ..$ rand64 : num [1:190]
0.0476 0.1 0.1556 0.1047 0.125 ... ..$ rand65 : num [1:190] 0.131 0.0375 0.0778 0.0698 0.0208
... ..$ rand66 : num [1:190] 0.25 0.25 0.139 0.18 0.271 ... ..$ rand67 : num [1:190] 0.0357
0.1375 0.0667 0.0698 0 ... ..$ rand68 : num [1:190] 0.1071 0.0833 0.1056 0.1298 0.1875 ... ..
..$ rand69 : num [1:190] 0.234 0.164 0.172 0.185 0.181 ... ..$ rand70 : num [1:190] 0.226
0.204 0.233 0.19 0.198 ... ..$ rand71 : num [1:190] 0.0714 0.1125 0.0444 0.093 0.1146 ... ..
..$ rand72 : num [1:190] 0.1508 0.0889 0.1611 0.1156 0.1181 ... ..$ rand73 : num [1:190]
0.175 0.226 0.106 0.127 0.191 ... ..$ rand74 : num [1:190] 0.0357 0.125 0.1556 0.0233 0.0729
... ..$ rand75 : num [1:190] 0.159 0.14 0.139 0.206 0.309 ... ..$ rand76 : num [1:190] 0
0.158 0.106 0.165 0.146 ... ..$ rand77 : num [1:190] 0.123 0.161 0.122 0.169 0.163 ... ..$
rand78 : num [1:190] 0.0635 0.1611 0.1444 0.2506 0.2153 ... ..$ rand79 : num [1:190] 0.0238
0.075 0.0556 0.093 0.1042 ... ..$ rand80 : num [1:190] 0.155 0.246 0.106 0.095 0.198 ... ..
..$ rand81 : num [1:190] 0.1706 0.0861 0.1889 0.1227 0.1944 ... ..$ rand82 : num [1:190]
0.262 0.154 0.289 0.202 0.135 ... ..$ rand83 : num [1:190] 0.127 0.0403 0.1278 0.1208 0.0972
... ..$ rand84 : num [1:190] 0.00397 0.05556 0.11111 0.07881 0.07639 ... ..$ rand85 : num
[1:190] 0.313 0.278 0.211 0.15 0.267 ... ..$ rand86 : num [1:190] 0.23 0.136 0.133 0.204 0.111
... ..$ rand87 : num [1:190] 0 0.1875 0.1444 0.1047 0.0833 ... ..$ rand88 : num [1:190]
0.2143 0.125 0.0333 0.1047 0.1771 ... ..$ rand89 : num [1:190] 0.333 0.358 0.267 0.403 0.312
... ..$ rand90 : num [1:190] 0.23 0.194 0.261 0.218 0.174 ... ..$ rand91 : num [1:190] 0.179
0.167 0.256 0.202 0.167 ... ..$ rand92 : num [1:190] 0.0357 0.0375 0.0556 0.1512 0.0729 ... ..
..$ rand93 : num [1:190] 0.0238 0.1625 0.1667 0.1628 0.0833 ... ..$ rand94 : num [1:190]
0.131 0.1875 0.0444 0.093 0.0625 ... ..$ rand95 : num [1:190] 0.1032 0.1181 0.1222 0.0556
0.1076 ... ..$ rand96 : num [1:190] 0.0556 0.0931 0.0778 0.1602 0.1701 ... ..$ rand97 :
num [1:190] 0.1825 0.1111 0.0778 0.0762 0.059 ... ..$ [list output truncated] ..$ : 'data.frame':
190 obs. of 102 variables: ..$ name1 : Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16
17 18 19 1 2 ... ..$ name2 : Factor w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... ..
..$ rand1 : num [1:190] 0.1151 0.0306 0.0778 0.137 0.1076 ... ..$ rand2 : num [1:190] 0.0397
0.0903 0.0611 0.1324 0.0903 ... ..$ rand3 : num [1:190] 0.1032 0.1181 0.0778 0.1137 0.0868
... ..$ rand4 : num [1:190] 0.0635 0.0403 0.0389 0.0627 0.0694 ... ..$ rand5 : num [1:190]
0.0833 0 0.0778 0.1047 0.0417 ... ..$ rand6 : num [1:190] 0.0238 0.05 0.1222 0.0465 0.0417 ...
.. ..$ rand7 : num [1:190] 0.167 0.146 0.106 0.153 0.104 ... ..$ rand8 : num [1:190] 0.0476
0.0375 0.0556 0.0116 0.0312 ... ..$ rand9 : num [1:190] 0.0516 0.0556 0.1 0.0672 0.0868 ...
.. ..$ rand10 : num [1:190] 0.0794 0.0556 0.0444 0.0672 0.066 ... ..$ rand11 : num [1:190]
0.0278 0.0278 0.1278 0.1092 0.059 ... ..$ rand12 : num [1:190] 0.0595 0.1958 0.1167 0.1415

```

0.0729\$ rand13 : num [1:190] 0.262 0.167 0.156 0.248 0.281\$ rand14 : num
[1:190] 0.0992 0.0861 0.0889 0.0762 0.0694\$ rand15 : num [1:190] 0.1706 0.1361 0.0778
0.146 0.1111\$ rand16 : num [1:190] 0.1349 0.1153 0.05 0.0743 0.0486\$ rand17
: num [1:190] 0.0952 0.0833 0.0722 0.095 0.1042\$ rand18 : num [1:190] 0.0833 0.025 0
0.0698 0.0729\$ rand19 : num [1:190] 0.0119 0.075 0.0333 0.1279 0.0625\$ rand20
: num [1:190] 0.0595 0.0125 0.0889 0.0814 0.0312\$ rand21 : num [1:190] 0.0675 0.0806
0.1222 0.1486 0.0347\$ rand22 : num [1:190] 0.0278 0.0403 0.0278 0.1092 0.1111
..\$ rand23 : num [1:190] 0.0714 0.0625 0.0333 0.0581 0.0417\$ rand24 : num [1:190]
0.0357 0.05 0.0222 0.0233 0.0625\$ rand25 : num [1:190] 0.0119 0.0375 0.0333 0.0698
0.0729\$ rand26 : num [1:190] 0.127 0.1181 0.1 0.0323 0.0972\$ rand27 : num
[1:190] 0.1429 0.0958 0.1056 0.0975 0.1042\$ rand28 : num [1:190] 0.1071 0.0375 0.0444
0.0233 0.0417\$ rand29 : num [1:190] 0.23 0.219 0.139 0.206 0.122\$ rand30 :
num [1:190] 0.183 0.186 0.189 0.146 0.111\$ rand31 : num [1:190] 0.0476 0.2333 0.0722
0.1415 0.1563\$ rand32 : num [1:190] 0.187 0.247 0.256 0.234 0.253\$ rand33 :
num [1:190] 0.0595 0.0125 0.1 0.0233 0.0312\$ rand34 : num [1:190] 0.0238 0.05 0.0222
0.0465 0\$ rand35 : num [1:190] 0.0952 0.05 0.0667 0.0349 0.1042\$ rand36 :
num [1:190] 0.0476 0.075 0.0667 0.1047 0.1146\$ rand37 : num [1:190] 0.0476 0.1 0.1222
0.0698 0.0938\$ rand38 : num [1:190] 0.19 0.179 0.144 0.134 0.188\$ rand39 :
num [1:190] 0 0 0.0889 0.0233 0.0104\$ rand40 : num [1:190] 0.04365 0.06806 0.05556
0.09044 0.00347\$ rand41 : num [1:190] 0.0119 0.0375 0.0111 0.1047 0.0729\$
rand42 : num [1:190] 0.214 0.275 0.183 0.192 0.25\$ rand43 : num [1:190] 0.0913 0.0931
0.0556 0.137 0.1181\$ rand44 : num [1:190] 0.1587 0.0861 0.1111 0.0762 0.0694\$
rand45 : num [1:190] 0.143 0.167 0.189 0.178 0.26\$ rand46 : num [1:190] 0.0278 0.0403
0.0389 0.0627 0.0382\$ rand47 : num [1:190] 0.0357 0.0625 0.0222 0.1137 0.0208
..\$ rand48 : num [1:190] 0.131 0.0708 0.1611 0.095 0.1042\$ rand49 : num [1:190] 0 0.075
0.0111 0.0465 0\$ rand50 : num [1:190] 0.119 0 0.0222 0.093 0.0625\$ rand51
: num [1:190] 0.1071 0.0625 0.0222 0.0349 0\$ rand52 : num [1:190] 0.00794 0.10278
0.08333 0.06266 0.10069\$ rand53 : num [1:190] 0.123 0.0861 0.1222 0.1227 0.2569
..\$ rand54 : num [1:190] 0.1071 0.0681 0.0778 0.0439 0.1042\$ rand55 : num [1:190]
0.0357 0.0875 0.0889 0.0116 0.0208\$ rand56 : num [1:190] 0.0833 0.0458 0.0833 0.1298
0.0833\$ rand57 : num [1:190] 0.0595 0.0625 0.0667 0.0698 0.0521\$ rand58 : num
[1:190] 0.135 0.136 0.1 0.123 0.101\$ rand59 : num [1:190] 0.1071 0.0833 0.0944 0.095
0.0938\$ rand60 : num [1:190] 0.0119 0 0.0333 0.0581 0.0104\$ rand61 : num
[1:190] 0.1984 0.1514 0.0944 0.1389 0.1285\$ rand62 : num [1:190] 0.0833 0.0528 0.0944
0.1182 0.184\$ rand63 : num [1:190] 0.119 0.1083 0.1944 0.0368 0.0729\$ rand64
: num [1:190] 0.0556 0.0931 0.0556 0.0788 0.0868\$ rand65 : num [1:190] 0.0357 0.0875
0.0333 0.0465 0.0312\$ rand66 : num [1:190] 0.123 0.1528 0.1278 0.051 0.0278\$
rand67 : num [1:190] 0.0992 0.0528 0.1056 0.0743 0.1111\$ rand68 : num [1:190] 0.0119
0.025 0.0556 0.0349 0.0729\$ rand69 : num [1:190] 0.0754 0.1111 0.1556 0.1344 0.1007
... ..\$ rand70 : num [1:190] 0.0595 0.0375 0.1 0.0698 0.0417\$ rand71 : num [1:190]
0.0238 0.025 0.1 0.0233 0.0833\$ rand72 : num [1:190] 0.1468 0.1111 0.0778 0.0995
0.0799\$ rand73 : num [1:190] 0.0119 0.05 0 0.0116 0.125\$ rand74 : num [1:190]
0.0873 0.0528 0.05 0.1001 0.0139\$ rand75 : num [1:190] 0.0952 0.0833 0.0611 0.1066
0.0521\$ rand76 : num [1:190] 0.0714 0 0.1333 0.0581 0.0521\$ rand77 : num
[1:190] 0 0.0125 0 0.0581 0\$ rand78 : num [1:190] 0.0119 0.0125 0.0556 0 0.0208
..\$ rand79 : num [1:190] 0.119 0.075 0.0778 0.0698 0.0208\$ rand80 : num [1:190] 0.0992
0.0278 0.0389 0.0627 0.0174\$ rand81 : num [1:190] 0 0.0625 0.0556 0.1163 0.0521
..\$ rand82 : num [1:190] 0.0873 0.1486 0.1333 0.1111 0.1528\$ rand83 : num [1:190]

```

0.0476 0.05 0.0556 0.0581 0.0417 ... ..$ rand84 : num [1:190] 0.0833 0.0625 0 0.0116 0 ... ..
..$ rand85 : num [1:190] 0.0992 0.0403 0.1056 0.051 0.059 ... ..$ rand86 : num [1:190] 0.0952
0.1208 0.1167 0.0601 0.125 ... ..$ rand87 : num [1:190] 0.0357 0.0375 0.0667 0 0.0417 ... ..
..$ rand88 : num [1:190] 0.0952 0.075 0.1111 0.0116 0.0312 ... ..$ rand89 : num [1:190] 0.0833
0.1125 0.0667 0.0581 0.0312 ... ..$ rand90 : num [1:190] 0.1071 0.1125 0.0667 0.0814 0 ... ..
..$ rand91 : num [1:190] 0.0556 0.1181 0.1333 0.0672 0.0972 ... ..$ rand92 : num [1:190] 0
0.025 0.0778 0.0233 0.0625 ... ..$ rand93 : num [1:190] 0 0.0625 0.0222 0.0581 0.1354 ... ..
..$ rand94 : num [1:190] 0 0.025 0.0111 0.0581 0 ... ..$ rand95 : num [1:190] 0.1429 0.1083
0.0611 0.1647 0.1146 ... ..$ rand96 : num [1:190] 0.119 0.075 0.0667 0.0233 0.0938 ... ..$
rand97 : num [1:190] 0.0833 0 0.0333 0.0349 0 ... .. [list output truncated] ..$ `data.frame`:
190 obs. of 102 variables: ..$ name1 : Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16
17 18 19 1 2 ... ..$ name2 : Factor w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 ... ..
..$ rand1 : num [1:190] 0.639 0.526 0.511 0.563 0.535 ... ..$ rand2 : num [1:190] 0.603 0.572
0.683 0.561 0.618 ... ..$ rand3 : num [1:190] 0.742 0.557 0.833 0.607 0.542 ... ..$ rand4 :
num [1:190] 0.627 0.41 0.444 0.397 0.649 ... ..$ rand5 : num [1:190] 0.714 0.704 0.561 0.614
0.583 ... ..$ rand6 : num [1:190] 0.409 0.319 0.394 0.457 0.538 ... ..$ rand7 : num [1:190]
0.405 0.517 0.517 0.545 0.427 ... ..$ rand8 : num [1:190] 0.456 0.696 0.489 0.58 0.58 ... ..
..$ rand9 : num [1:190] 0.591 0.426 0.411 0.389 0.649 ... ..$ rand10 : num [1:190] 0.667 0.629
0.461 0.533 0.427 ... ..$ rand11 : num [1:190] 0.365 0.626 0.367 0.307 0.368 ... ..$ rand12
: num [1:190] 0.762 0.642 0.683 0.626 0.708 ... ..$ rand13 : num [1:190] 0.413 0.497 0.478
0.513 0.556 ... ..$ rand14 : num [1:190] 0.635 0.553 0.533 0.452 0.58 ... ..$ rand15 : num
[1:190] 0.488 0.458 0.622 0.554 0.604 ... ..$ rand16 : num [1:190] 0.587 0.632 0.644 0.561
0.726 ... ..$ rand17 : num [1:190] 0.337 0.532 0.483 0.45 0.455 ... ..$ rand18 : num [1:190]
0.484 0.585 0.583 0.647 0.649 ... ..$ rand19 : num [1:190] 0.631 0.662 0.5 0.453 0.802 ... ..
..$ rand20 : num [1:190] 0.603 0.406 0.389 0.513 0.649 ... ..$ rand21 : num [1:190] 0.762
0.596 0.6 0.612 0.729 ... ..$ rand22 : num [1:190] 0.544 0.45 0.628 0.495 0.597 ... ..$
rand23 : num [1:190] 0.393 0.479 0.394 0.417 0.406 ... ..$ rand24 : num [1:190] 0.679 0.696
0.644 0.554 0.49 ... ..$ rand25 : num [1:190] 0.79 0.487 0.6 0.558 0.677 ... ..$ rand26 :
num [1:190] 0.484 0.547 0.411 0.594 0.389 ... ..$ rand27 : num [1:190] 0.44 0.417 0.583 0.48
0.521 ... ..$ rand28 : num [1:190] 0.27 0.522 0.322 0.455 0.333 ... ..$ rand29 : num [1:190]
0.389 0.606 0.472 0.492 0.42 ... ..$ rand30 : num [1:190] 0.702 0.683 0.567 0.627 0.479 ... ..
..$ rand31 : num [1:190] 0.476 0.442 0.361 0.364 0.594 ... ..$ rand32 : num [1:190] 0.516
0.59 0.489 0.499 0.497 ... ..$ rand33 : num [1:190] 0.643 0.586 0.339 0.591 0.427 ... ..$
rand34 : num [1:190] 0.5 0.492 0.433 0.563 0.639 ... ..$ rand35 : num [1:190] 0.317 0.551
0.678 0.528 0.326 ... ..$ rand36 : num [1:190] 0.694 0.669 0.344 0.5 0.604 ... ..$ rand37
: num [1:190] 0.623 0.632 0.567 0.596 0.663 ... ..$ rand38 : num [1:190] 0.393 0.621 0.489
0.45 0.583 ... ..$ rand39 : num [1:190] 0.389 0.413 0.494 0.657 0.58 ... ..$ rand40 : num
[1:190] 0.425 0.535 0.467 0.513 0.462 ... ..$ rand41 : num [1:190] 0.623 0.713 0.478 0.558
0.625 ... ..$ rand42 : num [1:190] 0.452 0.575 0.417 0.576 0.427 ... ..$ rand43 : num
[1:190] 0.452 0.482 0.6 0.514 0.434 ... ..$ rand44 : num [1:190] 0.389 0.289 0.656 0.54 0.701
... ..$ rand45 : num [1:190] 0.738 0.458 0.622 0.671 0.542 ... ..$ rand46 : num [1:190]
0.528 0.474 0.628 0.663 0.569 ... ..$ rand47 : num [1:190] 0.52 0.576 0.544 0.494 0.681 ... ..
..$ rand48 : num [1:190] 0.464 0.642 0.539 0.521 0.406 ... ..$ rand49 : num [1:190] 0.579
0.925 0.611 0.767 0.615 ... ..$ rand50 : num [1:190] 0.552 0.511 0.394 0.431 0.215 ... ..$
rand51 : num [1:190] 0.69 0.812 0.811 0.826 0.812 ... ..$ rand52 : num [1:190] 0.413 0.747
0.561 0.542 0.378 ... ..$ rand53 : num [1:190] 0.567 0.714 0.544 0.366 0.514 ... ..$ rand54
: num [1:190] 0.635 0.446 0.656 0.654 0.469 ... ..$ rand55 : num [1:190] 0.655 0.546 0.656
0.473 0.469 ... ..$ rand56 : num [1:190] 0.615 0.506 0.572 0.631 0.628 ... ..$ rand57 : num

```

```
[1:190] 0.48 0.708 0.444 0.572 0.601 ... .. .$ rand58 : num [1:190] 0.472 0.601 0.544 0.61 0.618
... .. .$ rand59 : num [1:190] 0.496 0.501 0.483 0.399 0.264 ... .. .$ rand60 : num [1:190]
0.552 0.611 0.406 0.617 0.674 ... .. .$ rand61 : num [1:190] 0.516 0.536 0.406 0.605 0.538 ... ..
.. .$ rand62 : num [1:190] 0.603 0.626 0.533 0.563 0.556 ... .. .$ rand63 : num [1:190] 0.484
0.514 0.567 0.726 0.41 ... .. .$ rand64 : num [1:190] 0.468 0.357 0.433 0.514 0.559 ... .. .$
rand65 : num [1:190] 0.548 0.625 0.511 0.628 0.573 ... .. .$ rand66 : num [1:190] 0.46 0.272
0.511 0.443 0.535 ... .. .$ rand67 : num [1:190] 0.627 0.685 0.606 0.582 0.611 ... .. .$ rand68
: num [1:190] 0.476 0.617 0.439 0.486 0.531 ... .. .$ rand69 : num [1:190] 0.429 0.375 0.428
0.494 0.51 ... .. .$ rand70 : num [1:190] 0.476 0.458 0.5 0.554 0.531 ... .. .$ rand71 : num
[1:190] 0.683 0.751 0.367 0.628 0.483 ... .. .$ rand72 : num [1:190] 0.44 0.25 0.517 0.483 0.531
... .. .$ rand73 : num [1:190] 0.552 0.557 0.361 0.443 0.392 ... .. .$ rand74 : num [1:190]
0.496 0.767 0.506 0.635 0.524 ... .. .$ rand75 : num [1:190] 0.46 0.46 0.6 0.571 0.368 ... .. .$
rand76 : num [1:190] 0.643 0.567 0.639 0.498 0.594 ... .. .$ rand77 : num [1:190] 0.56 0.596
0.578 0.531 0.385 ... .. .$ rand78 : num [1:190] 0.615 0.601 0.522 0.578 0.542 ... .. .$ rand79
: num [1:190] 0.595 0.583 0.511 0.651 0.667 ... .. .$ rand80 : num [1:190] 0.675 0.551 0.511
0.633 0.472 ... .. .$ rand81 : num [1:190] 0.425 0.576 0.556 0.459 0.399 ... .. .$ rand82 : num
[1:190] 0.579 0.597 0.511 0.548 0.545 ... .. .$ rand83 : num [1:190] 0.437 0.585 0.572 0.635
0.618 ... .. .$ rand84 : num [1:190] 0.611 0.432 0.733 0.63 0.569 ... .. .$ rand85 : num [1:190]
0.587 0.507 0.461 0.52 0.382 ... .. .$ rand86 : num [1:190] 0.413 0.393 0.55 0.48 0.535 ... ..
.. .$ rand87 : num [1:190] 0.472 0.45 0.522 0.616 0.493 ... .. .$ rand88 : num [1:190] 0.595 0.675
0.489 0.442 0.604 ... .. .$ rand89 : num [1:190] 0.583 0.454 0.489 0.516 0.552 ... .. .$ rand90
: num [1:190] 0.433 0.593 0.606 0.488 0.535 ... .. .$ rand91 : num [1:190] 0.353 0.615 0.567
0.429 0.66 ... .. .$ rand92 : num [1:190] 0.702 0.562 0.511 0.547 0.552 ... .. .$ rand93 : num
[1:190] 0.548 0.65 0.5 0.523 0.559 ... .. .$ rand94 : num [1:190] 0.536 0.487 0.878 0.57 0.812 ...
.. .. .$ rand95 : num [1:190] 0.635 0.674 0.617 0.64 0.715 ... .. .$ rand96 : num [1:190] 0.444
0.54 0.344 0.584 0.528 ... .. .$ rand97 : num [1:190] 0.409 0.428 0.378 0.517 0.33 ... .. [list
output truncated] $ special.crct:List of 1 .. $ SigbMPDi:List of 3 .. $ special.ses :'data.frame': 190
obs. of 5 variables: .. .. $ name1: Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18
19 1 2 ... .. $ name2: Factor w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 1 ... .. $ bin1
: num [1:190] 0 0 0 0 0 0 0 0 0 ... .. $ bin2 : num [1:190] 0 0 0 0 0 0 0 -99 0 0 ... .. $
bin3 : num [1:190] 0 0 0 0 0 0 0 0 0 ... .. $ special.rc :'data.frame': 190 obs. of 5 variables: .. ..
.. $ name1: Factor w/ 19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... .. $ name2:
Factor w/ 19 levels "s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 1 ... .. $ bin1 : num [1:190] 0 0 0 0 0
0 0 0 0 0 ... .. $ bin2 : num [1:190] 0 0 0 0 0 0 0 -1.1 0 0 ... .. $ bin3 : num [1:190] 0 0 0
0 0 0 0 0 ... .. $ special.conf:'data.frame': 190 obs. of 5 variables: .. .. $ name1: Factor w/
19 levels "s10","s11","s12",...: 11 13 14 15 16 17 18 19 1 2 ... .. $ name2: Factor w/ 19 levels
"s1","s10","s11",...: 1 1 1 1 1 1 1 1 1 1 ... .. $ bin1 : num [1:190] 0 0 0 0 0 0 0 0 0 ... .. $
bin2 : num [1:190] 0 0 0 0 0 0 0 -1.1 0 0 ... .. $ bin3 : num [1:190] 0 0 0 0 0 0 0 0 0 ...
```

Details

See help of icamp.big for detail.

Source

icamp.big result from the example.data.

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

Examples

```
data(icamp.out)
```

match.2col	<i>Check the consistency of the first two columns of different matrixes</i>
------------	---

Description

This function is usually used to check the consistency of samples names in different pairwise comparison matrixes.

Usage

```
match.2col(check.list, name.check = NULL, rerank = TRUE, silent = FALSE)
```

Arguments

check.list	List, each element is a matrix. It must be set in a format like "check.list=list(A=A,B=B)". The first two columns of the matrixes will be compared and matched with each other.
name.check	matrix, the first two columns will be used as a standard. The pairs not appear in this matrix will be removed from all matrixes.
rerank	Logic, make the first two columns in all matrixes in the same rank or not. Default is TRUE.
silent	Logic, whether to show messages. Default is FALSE, thus all messages will be showed.

Details

A tool to match IDs.

Value

Return a list object, new matrixes with the same first two columns. Some messages will return if some names are removed or all names matches very well.

Note

Version 2: 2020.8.19, add example. Version 1: 2018.10.20

Author(s)

Daliang Ning

Examples

```
# here two simple matrixes are generated and the pairwise comparison IDs not matched are removed.
A=1:5
names(A)=paste0("S",1:5)
B=1:6
names(B)=paste0("S",1:6)
DA3c=dist.3col(dist(A))
DB3c=dist.3col(dist(B))

checkid=match.2col(check.list = list(DA3c=DA3c,DB3c=DB3c))
DA3cnew=checkid$DA3c
DB3cnew=checkid$DB3c
```

match.name

*Check and ensure the consistency of IDs in different objects.***Description**

This function is usually used to check the consistency of species or samples names in different data table (e.g. OTU table and phylogenetic distance matrix). it can be used to check row names and/or column names of different matrixes, names in vector(s) or list(s), and tip.lable in tree(s)

Usage

```
match.name(name.check=integer(0), rn.list=list(integer(0)),
           cn.list=list(integer(0)), both.list=list(integer(0)),
           v.list=list(integer(0)), lf.list=list(integer(0)),
           tree.list=list(integer(0)), group=integer(0),
           rerank=TRUE, silent=FALSE)
```

Arguments

name.check	A character vector, indicating reference name list or the names you would like to keep. If not available, a union of all names is set as reference name list.
rn.list	A list object, including the matrix(es) of which the row names will be check. rn.list must be set in a format like "rn.list=list(A=A,B=B)". default is nothing.
cn.list	A list object, including the matrix(es) of which the column names will be check. cn.list must be set in a format like "cn.list=list(A=A,B=B)". default is nothing.
both.list	A list object, including the matrix(es) of which both column and row names will be check. both.list must be set in a format like "both.list=list(A=A,B=B)". default is nothing.
v.list	A list object, including the vector(s) of which the names will be check. v.list must be set in a format like "v.list=list(A=A,B=B)".default is nothing.

lf.list	A list object, including the list(s) of which the names will be check. lf.list must be set in a format like "lf.list=list(A=A,B=B)".default is nothing.
tree.list	A list object, including the tree(s) of which the tip.label names will be check. tree.list must be set in a format like "tree.list=list(A=A,B=B)".default is nothing.
group	a vector or one-column matrix/data.frame indicating the grouping information of samples or species, of which the sample/species names will be check.
rerank	Logic, make all names in the same rank or not. Default is TRUE
silent	Logic, whether to show messages. Default is FALSE, thus all messages will be showed.

Details

In many cases and functions, species names and samples names must be checked and set in the same rank. Sometimes, we also need to select some samples or species as necessary. This function can help.

Value

Return a list object, new matrixes with the same row/column names in the same rank. Some messages will return if some names are removed or all names match very well.

Note

Version 3: 2017.3.13 Version 2: 2015.9.25

Author(s)

Daliang Ning

Examples

```
data("example.data")
comm=example.data$comm
treat=example.data$treat
tree=example.data$tree
pd=example.data$pd
clas=example.data$classification

env=example.data$env
# remove one sample in purpose to see how match.name works
env=env[-13,]

sampid.check=match.name(rn.list = list(comm=comm, treat=treat, env=env))
comm.ck=sampid.check$comm
comm.ck=comm.ck[,colSums(comm.ck)>0,drop=FALSE]
treat.ck=sampid.check$treat
env.ck=sampid.check$env

taxid.check=match.name(cn.list = list(comm.ck=comm.ck),
                      rn.list = list(clas=clas),
```

```

                                tree.list = list(tree=tree))
comm.ck=taxid.check$comm.ck
clas.ck=taxid.check$clas
tree.ck=taxid.check$tree

```

maxbigm

Find maximum value in a big matrix

Description

Return the maximum value and its (their) location(s) in a big matrix.

Usage

```
maxbigm(m.desc, m.wd, nworker = 1, rm.na = TRUE, size.limit = 10000 * 10000)
```

Arguments

m.desc	the name of the file to hold the backingfile description of the big matrix.
m.wd	the path of the folder holding the big matrix file.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
rm.na	logic, whether to remove NA. Default is TRUE.
size.limit	the matrix size which your current computer memory can easily handle at each time.

Details

A tool to figure out the maximum value in the big phylogenetic distance matrix.

Value

Output is a list of two elements.

max.value	Numeric, the maximum value.
row.col	Matrix, the row(s) and column(s), i.e. the location(s), of the maximum value in the big matrix.

Note

Version 3: 2020.9.1, remove setwd; change dontrun to donttest and revise save.wd in help doc.
Version 2: 2020.8.19, add example. Version 1: 2015.12.16

Author(s)

Daliang Ning

References

Michael J. Kane, John Emerson, Stephen Weston (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[midpoint.root.big](#)

Examples

```
# this example shows how to find maximum value
# in a big phylogenetic distance matrix.
data("example.data")
tree=example.data$tree
# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)

maxb=maxbigm(m.desc = pd.big$pd.file, m.wd = pd.big$pd.wd,
             nworker = nworker, rm.na = TRUE)
```

midpoint.root.big *Midpoint root a large phylogeny*

Description

This is modified from the function "modpoint.root" in package "phytools". To deal with a large tree, phylogenetic distance is calculated and saved by using bigmemory in advance.

Usage

```
midpoint.root.big(tree, pd.desc, pd.spname, pd.wd, nworker = 4)
```

Arguments

tree	phylogenetic tree, an object of class "phylo".
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemmory file of the phylogenetic distance matrix are saved.

nworker integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.

Details

iCAMP analysis need a rooted tree. If it is difficult to figure out the root, midpoint root is recommended for iCAMP analysis. Modified from the function 'midpoint.root' in package 'phytool' (Revell 2012), this function uses bigmemory (Kane et al 2013) to deal with large datasets.

Value

Output is a list with two elements.

tree The rooted tree.
max.pd The maximum pairwise phylogenetic distance.

Note

Version 3: 2020.9.1, remove setwd; change dontrun to donttest and revise save.wd in help doc.
Version 2: 2020.8.19, update help document, add example. Version 1: 2015.12.16

Author(s)

Daliang Ning

References

Farris, J. (1972) Estimating phylogenetic trees from distance matrices. *American Naturalist*, 106, 645-667.
Paradis, E., J. Claude, and K. Strimmer (2004) APE: Analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20, 289-290.
Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, 3, 217-223.
Michael J. Kane, John Emerson, Stephen Weston (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[maxbigm](#)

Examples

```
data("example.data")
tree=example.data$tree
# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.
```

```

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)

mroot=midpoint.root.big(tree = tree, pd.desc = pd.big$pd.file,
                        pd.spname = pd.big$tip.label,
                        pd.wd = pd.big$pd.wd, nworker = nworker)

```

mntdn

Mean nearest taxon distance (MNTD)

Description

Calculate mean nearest taxon distance (MNTD) in each community in a given community matrix.

Usage

```

mntdn(comm, pd, abundance.weighted = TRUE,
       check.name = TRUE, memory.G = 50, time.count = FALSE)

```

Arguments

comm	matrix or data.frame, community data matrix, rownames are sample names, colnames are OTU ids.
pd	matrix, pairwise phylogenetic distance matrix.
abundance.weighted	logic, whether weighted by species abundance, default is TRUE, means weighted.
check.name	logic, whether to check the OTU ids (species names) in community matrix and phylogenetic distance matrix are the same.
memory.G	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb
time.count	logic, whether to count calculation time, default is FALSE.

Details

mean nearest taxon distance (MNTD) in each community, using the same algorithm as the function 'mntd' in package 'picante'.

Value

result is a numeric vector with sample names

Note

Version 2: 2020.8.19, update help document, add example. Version 1: 2017.3.13

Author(s)

Daliang Ning

References

Webb CO, Ackerly DD, and Kembel SW. (2008). Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics* 18:2098-2100

Kembel, S.W., Cowan, P.D., Helmus, M.R., Cornwell, W.K., Morlon, H., Ackerly, D.D. et al. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463-1464.

See Also[NTI.p](#)**Examples**

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
mntd=mntdn(comm=comm,pd=pd,abundance.weighted = TRUE)
```

mpdn

Mean pairwise distance (MPD)

Description

Calculate mean pairwise distance (MPD) in each community in a given community matrix.

Usage

```
mpdn(comm, pd, abundance.weighted = TRUE, time.output = FALSE)
```

Arguments

comm	matrix or data.frame, community data matrix, rownames are sample names, colnames are OTU ids.
pd	matrix, pairwise phylogenetic distance matrix.
abundance.weighted	logic, whether weighted by species abundance, default is TRUE, means weighted.
time.output	logic, whether to count calculation time, default is FALSE.

Details

mean pairwise distance (MPD) in each community, which is the same index as 'mpd' in package 'picante', but calculated by matrix multiplication.

Value

result is a numeric vector with sample names

Note

Version 2: 2020.8.19, update help document, add example. Version 1: 2017.3.13

Author(s)

Daliang Ning

References

Webb C, Ackerly D, McPeck M, and Donoghue M. (2002). Phylogenies and community ecology. *Annual Review of Ecology and Systematics* 33:475-505.

Kembel, S.W., Cowan, P.D., Helmus, M.R., Cornwell, W.K., Morlon, H., Ackerly, D.D. et al. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463-1464.

See Also

[NRI.p](#)

Examples

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
mpd=mpdn(comm = comm, pd = pd, abundance.weighted = TRUE)
```

NRI.p

Calculate net relatedness index (NRI) by parallel computing.

Description

Calculate net relatedness index (NRI) or other index of null model significance test based on mean pairwise distance (MPD) by parallel computing, for small and medium size dataset.

Usage

```
NRI.p(comm, dis, nworker = 4, memo.size.GB = 50,
       weighted = c(TRUE, FALSE), check.name = TRUE,
       rand = 1000, output.MPD = c(FALSE, TRUE), silent = FALSE,
       sig.index=c("SES", "NRI", "Confidence", "RC", "all"))
```


Arguments

<code>comm</code>	community data matrix. rownames are sample names. colnames are species names.
<code>dis</code>	Phylogenetic distance matrix
<code>nworker</code>	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
<code>memo.size.GB</code>	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
<code>weighted</code>	Logic, consider abundances or not (just presence/absence). default is TRUE.
<code>check.name</code>	Logic, whether to check the species names in comm and dis. default is TRUE.
<code>rand</code>	integer, randomization times. default is 1000.
<code>output.MPD</code>	Logic, whether to output observed MNTD, so that you do not need to calculate observed MNTD alone. default is FALSE.
<code>silent</code>	Logic, if FALSE, some messages will be showed during calculation. Default is FALSE.
<code>sig.index</code>	character, the index for null model significance test. SES or NRI, standard effect size, i.e. net relatedness index (NRI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on MPD, i.e. count the number of null MPD lower than observed MPD plus a half of the number of null MPD equal to observed MPD, to get alpha, then calculate MPD-based RC as $(2 \times \alpha - 1)$; all, output all the three indexes. default is SES. If input a vector, only the first element will be used.

Details

The net relatedness index (NRI) is a standardized measure of the mean pairwise phylogenetic distance in each sample/community (MPD). Currently this function only performs one null model algorithm, "taxa.labels" ("taxa shuffle", Kembel 2009), which is to shuffle distance matrix labels (across all taxa included in distance matrix). If the randomized results are all the same, the standard deviation will be zero and NRI will be NAN. In this case, NRI will be set as zero, since the observed result is not differentiable from randomized results.

RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed diversity index deviates from null expectation, which could be a better metric than standardized effect size (NRI) in some cases, e.g. null values do not follow normal distribution.

Value

Output can be a data.frame with each row representing a sample and only one column of index values, or a list of several data.frame objects.

SES output if sig.index is Confidence or all, a data.frame with NRI value for each sample.

Confidence	output if sig.index is SES or all, a data.frame showing confidence level based on MPD for each sample.
RC	output if sig.index is RC or all, a data.frame showing RC based on MPD for each sample.
MPD.obs	output if output.MPD is TRUE, a data.frame showing observed MPD for each sample.
MPD.rand	output if output.MPD is TRUE, a matrix showing all null MPD values.

Note

Version 2: 2020.8.19, update help document, add example Version 1: 2017.5.10

Author(s)

Daliang Ning

References

Webb CO, Ackerly DD, and Kembel SW. 2008. Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics* 18:2098-2100

Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.

Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[mpdn](#)

Examples

```
data("example.data")
comm=example.data$comm
pd=example.data$pd
nworker=2 # parallel computing thread number.
rand.time=20 # usually use 1000 for real data.
sigmpd=NRI.p(comm=comm, dis=pd, nworker=nworker,
             weighted=TRUE, rand=rand.time,
             sig.index="all")
NRI=sigmpd$SES
CMPD=sigmpd$Confidence
RCMPD=sigmpd$RC
```

NTI.p

Calculate nearest taxon index (NTI) with parallel computing

Description

Calculate nearest taxon index (NTI) of each sample with parallel computing.

Usage

```
NTI.p(comm, dis, nworker = 4, memo.size.GB = 50,
      weighted = c(TRUE, FALSE), rand = 1000,
      check.name = TRUE, output.MNTD = c(FALSE, TRUE),
      sig.index=c("SES", "NTI", "Confidence", "RC", "all"),
      silent=FALSE)
```

Arguments

comm	community data matrix. rownames are sample names. colnames are species names.
dis	Phylogenetic distance matrix.
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memo.size.GB	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
weighted	Logic, consider abundances or not (just presence/absence). default is TRUE.
rand	integer, randomization times. default is 1000.
check.name	logic, whether to check the taxa names in comm and dis, which must be the same and in the same order; if not match, remove mismatched names and change to the same order. default is TRUE.
output.MNTD	logic, if TRUE, the NTI and MNTD will be output, if FALSE, only output NTI.
sig.index	character, the index for null model significance test. SES or NTI, standard effect size, i.e. nearest taxon index (NTI); Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level; RC, modified Raup-Crick index (RC) based on MNTD, i.e. count the number of null MNTD lower than observed MNTD plus a half of the number of null MNTD equal to observed MNTD, to get alpha, then calculate MNTD-based RC as $(2 \times \alpha - 1)$; all, output all the three indexes. default is SES. If input a vector, only the first element will be used.
silent	logic, if FALSE, some messages will show during calculation.

Details

The nearest taxon index (NTI) is a standardized measure of the mean phylogenetic distance to the nearest taxon in each sample/community (MNTD). Currently this function only performs one null model algorithm, "taxa.labels" ("taxa shuffle", Kembel 2009), which is to shuffle distance matrix labels (across all taxa included in distance matrix). If the randomized results are all the same, the standard deviation will be zero and NTI will be NAN. In this case, NTI will be set as zero, since the observed result is not differentiable from randomized results.

RC (Chase et al 2011) and Confidence (Ning et al 2020) are alternative significance test indexes to evaluate how the observed diversity index deviates from null expectation, which could be a better metric than standardized effect size (NTI) in some cases, e.g. null values do not follow normal distribution.

Value

If output.MNTD is FALSE, output is a one-column matrix where rownames are sample IDs and the only column shows NTI values. If output.MNTD is TRUE, output is a list of three elements.

NTI	matrix, NTI values.
MNTD	matrix, observed MNTD.
MNTD.rand	array, null MNTD values, the third dimension represent randomization times.

Note

Version 2: 2020.8.19, update help document, add example. Version 1: 2018.10.19

Author(s)

Daliang Ning

References

- Webb CO, Ackerly DD, and Kembel SW. 2008. Phylocom: software for the analysis of phylogenetic community structure and trait evolution. *Bioinformatics* 18:2098-2100
- Kembel, S.W. (2009). Disentangling niche and neutral influences on community assembly: assessing the performance of community phylogenetic structure tests. *Ecol Lett*, 12, 949-960.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.
- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

See Also

[mntdn](#)

Examples

```

data("example.data")
comm=example.data$comm
pd=example.data$pd
nworker=2 # parallel computing thread number.
rand.time=4 # usually use 1000 for real data.
sigmntd=NTI.p(comm=comm, dis=pd, nworker = nworker,
              weighted = TRUE, rand = rand.time,
              sig.index="all")
NTI=sigmntd$SES
CMNTD=sigmntd$Confidence
RCMNTD=sigmntd$RC

```

null.norm

Normality test for null values

Description

To test whether the null values of each turnover of each bin follow normal distribution.

Usage

```

null.norm(icamp.output = NULL, rand.list = NULL, index.name = "Test.Index",
          p.norm.cut = 0.05, detail.out = FALSE)

```

Arguments

icamp.output	list, the exact output of the function icamp.big in which detail.null must be TRUE, to save all null values.
rand.list	list, the null values of a certain dissimilarity index. Each element is a matrix that represents a bin. In each element matrix, the first two columns indicate sample IDs of the pairwise comparison (turnover), and each of the other columns shows the null values from one time of randomization.
index.name	character, when rand.list is given, to specify the name of the dissimilarity index.
p.norm.cut	numeric, the threshold of significant P value. A p value lower than this indicates significant difference from normal distribution.
detail.out	logic, if TRUE, the detailed statistics and P values for each turnover of each bin will be output; otherwise, only output a summary on non-normal percentage for each bin.

Details

Normal distribution of null values is basic assumption when using Standard Effect Size (SES, e.g. betaNRI, betaNTI) to identify significant difference between null and observed values. This function uses five different methods to perform normality test, including Anderson-Darling test (Anderson), Cramer-von Mises test (Cramer), Kolmogorov-Smirnov test (Kolmogorov, also known as Lilliefors test), Shapiro-Francia test (ShapiroF), and Shapiro-Wilk test (Shapiro). The function 'shapiro.test' in package 'stats', and various functions in package 'nortest' are used.

Value

Output is a list object.

summary	data.frame, each row represents a bin and a dissimilarity index. Seven columns. The first column indicates the dissimilarity index; the second column indicate Bin ID; each of the other columns indicate non-normal ratio based on a method. The non-normal ratio is calculated as percentage of turnovers where null value distribution is significantly different from normal distribution.
P.value.cut	the value of p.norm.cut
detail	list, each first-level element represents a dissimilarity index; each second-level element is a matrix represents a bin; and the matrix has 14 columns, including the dissimilarity index (Index), bin ID (BinID), sample IDs (name1 and name2), and the statistics and P value based on different methods.

Note

Version 2: 2020.8.19, update help document, add example Version 1: 2020.8.1

Author(s)

Daliang Ning

References

- Stephens, M.A. (1986): Tests based on EDF statistics. In: D'Agostino, R.B. and Stephens, M.A., eds.: Goodness-of-Fit Techniques. Marcel Dekker, New York.
- Dallal, G.E. and Wilkinson, L. (1986): An analytic approximation to the distribution of Lilliefors' test for normality. *The American Statistician*, 40, 294-296.
- Stephens, M.A. (1974): EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69, 730-737.
- Royston, P. (1993): A pocket-calculator algorithm for the Shapiro-Francia test for non-normality: an application to medicine. *Statistics in Medicine*, 12, 181-184.
- Thode Jr., H.C. (2002): *Testing for Normality*. Marcel Dekker, New York
- Patrick Royston (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31, 115-124. doi: 10.2307/2347973.
- Patrick Royston (1982). Algorithm AS 181: The W test for Normality. *Applied Statistics*, 31, 176-180. doi: 10.2307/2347986.
- Patrick Royston (1995). Remark AS R94: A remark on Algorithm AS 181: The W test for normality. *Applied Statistics*, 44, 547-551. doi: 10.2307/2986146.
- Juergen Gross and Uwe Ligges (2015). nortest: Tests for Normality. R package version 1.0-4. <https://CRAN.R-project.org/package=nortest>

See Also

[icamp.big](#), [change.sigindex](#)

Examples

```
data("icamp.out")
nntest=null.norm(icamp.output = icamp.out, detail.out = TRUE)
```

pdist.big

*Pairwise phylogenetic distance matrix from big tree***Description**

Calculates between-species phylogenetic distance matrix from a tree, using bigmemory to deal with too large dataset.

Usage

```
pdist.big(tree, wd = getwd(), tree.asbig = FALSE,
          output = FALSE, nworker = 4, nworker.pd = nworker,
          memory.G = 50, time.count = FALSE,
          treepath.file="path.rda", pd.spname.file="pd.taxon.name.csv",
          pd.backingfile="pd.bin", pd.desc.file="pd.desc",
          tree.backingfile="treeinfo.bin", tree.desc.file="treeinfo.desc")
```

Arguments

tree	phylogenetic tree, an object of class "phylo".
wd	path of a folder to save the big phylogenetic distance matrix, default is current work directory.
tree.asbig	logic, whether to treat tree attributes also as big data, default is FALSE, generally no need to set as TRUE.
output	logic, whether to output the big phylogenetic distance matrix, default is FALSE, generally do not output it, could be too large.
nworker	for parallel computing the tree paths. a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
nworker.pd	for parallel computing the phylogenetic distance matrix. default is set the same as nworker. may need to set lower than nworker if the matrix is too large.
memory.G	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb
time.count	logic, whether to count calculation time, default is FALSE.
treepath.file	character, name of the file saving the tree.path, which is a list of all the nodes and edge lengths from root to every tip and/or node. it should be a .rda filename.
pd.spname.file	character, name of the file saving the taxa IDs, which has exactly the same order as the row names (and column names) of the big phylogenetic distance matrix. it should be a .csv filename.

<code>pd.backingfile</code>	character, the root name for the file for the cache of the big phylogenetic distance matrix. it should be a <code>.bin</code> filename.
<code>pd.desc.file</code>	character, name of the file to hold the backingfile description for the big phylogenetic distance matrix. it should be a <code>.desc</code> filename.
<code>tree.backingfile</code>	character, the root name for the file for the cache of the 3-column matrix of the tree information, including edge and edge length. it should be a <code>.bin</code> filename.
<code>tree.desc.file</code>	character, name of the file to hold the backingfile description for the tree information matrix. it should be a <code>.desc</code> filename.

Details

The cophenetic distance between each pair of taxa is calculated (Sokal and Rohlf 1962). Modified from the function "cophenetic" in package "ape" (Paradis & Schliep 2018), this function can calculate pairwise distance from large phylogenetic tree quickly by parallel computing. This function uses bigmemory (Kane et al 2013) to deal with large phylogenetic distance matrix, which will not occupy memory but directly be saved at the hard disk.

Value

Output is a list

<code>tip.label</code>	OTU ids or species names, which is <code>tip.label</code> in tree file.
<code>pd.wd</code>	the folder saving the big phylogenetic distance matrix.
<code>pd.file</code>	the folder saving the big phylogenetic distance matrix.
<code>pd.name.file</code>	the file saving the <code>tip.label</code> information.

Note

Version 4: 2020.9.1, remove `setwd`; add options to specify the file names; change `dontrun` to `donttest` and revise `save.wd` in help doc. Version 3: 2020.8.19, add example. Version 2: 2017.3.13 Version 1: 2015.7.24

Author(s)

Daliang Ning

References

- Sokal, R. R. & Rohlf, F. J.. (1962). The comparison of dendrograms by objective methods. *Taxon*, 11:33-40
- Paradis, E. & Schliep, K. (2018). *ape* 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 35: 526-528.
- Kane, M.J., Emerson, J., & Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

Examples

```

data("example.data")
tree=example.data$tree
# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the pd.big output.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)

```

ps.bin

*Test within-bin phylogenetic signal***Description**

Use Mantel test to evaluate phylogenetic signal within each bin, i.e. correlation between phylogenetic distance and niche difference.

Usage

```

ps.bin(sp.bin, sp.ra, sname.use = NULL,
       pd.desc = "pd.desc", pd.sname, pd.wd,
       nd.list, nd.sname = NULL, ndbig.wd = NULL,
       cor.method = c("pearson", "spearman"),
       r.cut = 0.01, p.cut = 0.2, min.spn = 6)

```

Arguments

sp.bin	one-column matrix or data.frame, indicating the bin ID for each species (OTU or ASV), rownames are species IDs. usually use the third column of "sp.bin" in the output of taxa.binphy.big . if input matrix with multiple columns, only the first column will be used.
sp.ra	one-column matrix or data.frame, or a vector with name for each element, indicating mean relative abundance of each species.
sname.use	character vector, to specify which species will be used for phylogenetic signal test. Default is NULL, means to use all species.
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.sname	character vector, species id in the same order as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.

nd.list	list object. if the niche difference matrixes are big.matrix, each element of this list is the big.matrix backingfile description, e.g. "pH.ND.desc"; otherwise, each element is a niche difference matrix based on an environment factor. usually this is the "nd" in the output of <code>dniche</code> .
nd.spname	character vector or NULL. If the niche difference matrixes are big.matrix, this is the species IDs in the same order as in each big matrix; otherwise, this should be set as NULL, the species IDs will be extracted from nd.list.
ndbig.wd	folder path or NULL. If the niche difference matrixes are big.matrix, this is where the big matrixes of niche differences are saved; otherwise, this is NULL.
cor.method	Correlation method, as accepted by cor: "pearson", "spearman" or "kendall". Multiple methods at a time are allowed.
r.cut	the cutoff of correlaiton coefficient to identify significant correlation.
p.cut	the cutoff of p value to identify significant correlation.
min.spn	the minimal species (or OUT or ASV) number required for phylogenetic signal test.

Details

This is simply Mantel test between phylogenetic distance and niche difference (i.e. phylogenetic signal) within each bin. Then, it returns the overall relative abundance of bins with significant phylogenetic signal, average correlation coefficient, as well as detailed results in each bin, to evaluate within-bin phylogenetic signal of the binning (inputed as sp.bin). Bigmemory (Kane et al 2013) is used to deal with large datasets.

Value

Output is a list object with two elements.

Index	Summary of phylogenetic signal test. The indexes include relative abundance of bins with significant phylogenetic signal in all bins (RAsig) or in bins with species number larger than min.spn (RAsig.adj), average correlation coefficient in significant bins (MeanR.sig) or in all bins (MeanR).
detail	correlation coefficient (r) and p value in each bin.

Note

Version 3: 2020.9.1, remove setwd; change dontrun to donttest and revise save.wd in help doc.
Version 2: 2020.8.18, update help document, add example. Version 1: 2020.5.15

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

Kane, M.J., Emerson, J., & Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[taxa.binphy.big](#), [dniche](#)

Examples

```
data("example.data")
comm=example.data$comm
env=example.data$env
tree=example.data$tree

# since big.memory need to specify a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the big niche difference matrix.
nworker=2 # parallel computing thread number
niche.dif=dniche(env = env, comm = comm,
                method = "niche.value", nworker = nworker,
                out.dist=FALSE,bigmemo=TRUE,nd.wd = save.wd,
                nd.spname.file="nd.names.csv")
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)

ds = 0.2 # setting can be changed to explore the best choice
bin.size.limit = 5 # setting can be changed to explore the best choice.
# here, bin.size.limit is set as 5 just for the small example dataset.
# For real data, usually try 12 to 48.

phylobin=taxa.binphy.big(tree = tree, pd.desc = pd.big$pd.file,
                        pd.spname = pd.big$tip.label, pd.wd = pd.big$pd.wd,
                        ds = ds, bin.size.limit = bin.size.limit,
                        nworker = nworker)
sp.bin=phylobin$sp.bin[,3,drop=FALSE]

sp.ra=colMeans(comm/rowSums(comm))
abcut=3 # you may remove some species, if they are too rare to perform reliable correlation test.
commc=comm[,colSums(comm)>=abcut,drop=FALSE]
dim(commc)
spname.use=colnames(commc)

binps=ps.bin(sp.bin = sp.bin,sp.ra = sp.ra,spname.use = spname.use,
            pd.desc = pd.big$pd.file, pd.spname = pd.big$tip.label,
            pd.wd = pd.big$pd.wd, nd.list = niche.dif$nd,
```

```
nd.spname = niche.dif$names, ndbig.wd = niche.dif$nd.wd,
cor.method = "pearson", r.cut = 0.1, p.cut = 0.05, min.spn = 5)
```

qp.bin.js

*Calculate relative importance of community assembly processes***Description**

Identify the community assembly process governing each bin in each turnover (i.e. pairwise comparison between two samples/communities), then calculate the relative importance of community assembly processes in each turnover.

Usage

```
qp.bin.js(sig.phy.bin=NULL, sig.phy2.bin=NULL, sig.tax.bin=NULL,
          bin.weight, sig.phy.cut=1.96, sig.phy2.cut=1.96,
          sig.tax.cut=0.95, check.name=FALSE)
```

Arguments

sig.phy.bin	matrix, the first two columns are sample IDs, thus each row represent a turnover between two samples. From the third column, each column shows the significance testing index of phylogenetic null model analysis (e.g. betaNTI) of a bin. NULL means the data is not available.
sig.phy2.bin	matrix, the same as sig.phy.bin, serves as the second phylogenetic metrics when phylogenetic null model test is based on two different beta diversity indexes, e.g. both betaMPD and betaMNTD. NULL means the data is not available.
sig.tax.bin	matrix, the first two columns are sample IDs, thus each row represent a turnover between two samples. From the third column, each column shows the significance testing index of taxonomic null model analysis (e.g. RC.Bray) of a bin. NULL means the data is not available.
bin.weight	matrix, the first two columns are sample IDs, thus each row represent a turnover between two samples. From the third column, each column shows the abundance sum of a bin in each pair of samples.
sig.phy.cut	numeric, a cutoff for the null model significance testing index based on a phylogenetic beta diversity index, e.g. betaNRI based on betaMPD, default is 1.96.
sig.phy2.cut	numeric, a cutoff for the null model significance testing index based on the second phylogenetic beta diversity index, e.g. betaNTI based on betaMNTD, default is 1.96.
sig.tax.cut	numeric, a cutoff for the null model significance testing index based on a taxonomic beta diversity index, e.g. RC based on Bray-Curtis, default is 0.95.
check.name	logic, whether to check the sample IDs in different input matrixess are in the same order.

Details

The framework is proposed by James Stegen (2013 and 2015), to identify governing ecological process based on phylogenetic (betaNTI) and taxonomic (RC.Bray) null model analysis. In all pairwise comparisons between samples/communities, the non-random phylogenetic turnovers recognized by phylogeny shuffle were counted as influence of environment selection, and the non-random taxonomic turnovers in the rest pairwise comparisons were counted as influence of dispersal limitation or homogenizing dispersal. The rest part is called undominated.

This function applied this framework to each phylogenetic bin and allowed to use betaNRI and/or betaNTI. When both betaNTI and betaNRI are provided, a turnover is identified as controlled by selection when either betaNRI or betaNTI is significant. Alternatively, RC or confidence level based on betaMPD and/or betaMNTD can also be used (Ning et al. 2020).

Value

Output is a matrix. The first two columns are sample IDs, and each row represent a turnover between two samples. From the third column, each column shows the relative importance of a community assembly process in each turnover (pairwise comparison between each pair of samples).

Note

Version 5: 2020.8.19, update help document, add example. Version 4: 2020.7.28, change bNTI.bin=NULL, bNRI.bin=NULL to sig.phy.bin and sig.tax.bin. Version 3: 2018.10.20, add bNRI.bin as option; add check.name. Version 2: 2016.3.26, add RC.all option Version 1: 2015.12.16

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

Stegen JC, Lin X, Fredrickson JK, Chen X, Kennedy DW, Murray CJ et al. 2013. Quantifying community assembly processes and identifying features that impose them. *Isme Journal* 7: 2069-2079.

Stegen JC, Lin X, Fredrickson JK, Konopka AE. 2015. Estimating and mapping ecological processes influencing microbial community assembly. *Frontiers in Microbiology* 6.

See Also

[icamp.big](#)

Examples

```
data("icamp.out")
bNRIbins=icamp.out$detail$SigMPDi
RCbins=icamp.out$detail$SigBCa
binwt=icamp.out$detail$bin.weight
```

```
qpbins=qp.bin.js(sig.phy.bin = bNRIbins,sig.tax.bin = RCbins,
  bin.weight = binwt, sig.phy.cut = 1.96,
  sig.tax.cut = 0.95, check.name = TRUE)
```

qpen	<i>Quantifying assembly processes based on entire-community null model analysis</i>
------	---

Description

The previous framework quantifying assembly processes based on entire-community null model analysis (Stegen et al 2013, 2015). add bigmemory function to handle large datasets.

Usage

```
qpen(comm = NULL, pd = NULL, pd.big.wd = NULL,
  pd.big.spname = NULL, tree = NULL,
  bNTI = NULL, RC = NULL, ab.weight = TRUE,
  meta.ab = NULL, exclude.conspecifics = FALSE,
  rand.time = 1000, sig.bNTI = 1.96, sig.rc = 0.95,
  nworker = 4, memory.G = 50, project = NA,
  wd = getwd(), output.detail = FALSE, save.bNTIRC = FALSE)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a taxon (species or OTU or ASV), thus rownames should be sample IDs, colnames should be taxa IDs.
pd	a character or a matrix or NULL. If it is a character, it specifies the name of the file to hold the backingfile description of the big phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function. If it is matrix, it is the phylogenetic distance matrix with taxa IDs as colnames and rownames. The function will check the consistency of species names between phylogenetic matrix and comm. if pd is NULL, the function will calculate phylogenetic distance matrix from tree.
pd.big.wd	a folder path, where the bigmemory file of the phylogenetic distance matrix are saved. If pd has given the phylogenetic matrix, pd.big.wd should be NULL. If pd is NULL and pd.big.wd is NULL either, a folder will be created to save the big phylogenetic distance matrix.
pd.big.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances. not necessary if pd has given the phylogenetic matrix.
tree	phylogenetic tree, an object of class "phylo".
bNTI	a matrix, if beta nearest taxon index (betaNTI) values are available, just directly input them as a square matrix here.
RC	a matrix, if modified Raup-Crick metric based Bray-Curtis (RCbray) values are available, just directly input them as a square matrix here.

<code>ab.weight</code>	logic, abundance-weighted or binary. default is TRUE, means abundance-weighted.
<code>meta.ab</code>	a numeric vector, to define the relative abundance of each species in the regional pool. Default setting is NULL, means to calculate meta.ab as average relative abundance of each species across the samples.
<code>exclude.conspecific</code>	Logic, should conspecific taxa in different communities be exclude from MNTD calculations? default is FALSE. The same as in the function <code>bmntd</code> .
<code>rand.time</code>	integer, randomization times. default is 1000.
<code>sig.bNTI</code>	numeric, the cutoff of significant betaNTI, default is 1.96.
<code>sig.rc</code>	numeric, the cutoff of significant modified Raup-Crick metric, default is 0.95.
<code>nworker</code>	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
<code>memory.G</code>	numeric, to set the memory size as you need, so that calculation of large tree will not be limited by physical memory. unit is Gb. default is 50Gb.
<code>project</code>	character string, the prefix of saved output files.
<code>wd</code>	a folder path, where the files will be saved.
<code>output.detail</code>	logic, if TRUE, some detailed results including all null values will be output.
<code>save.bNTIRC</code>	logic, if TRUE, a file will be saved in the folder specified by <code>wd</code> .

Details

This method is developed by Dr. James Stegen (2013 and 2015), from which we developed iCAMP. In all pairwise comparisons between samples/communities, the non-random phylogenetic turnovers recognized by phylogeny shuffle were counted as influence of environment selection ($\beta\text{NTI} > 1.96$ or < -1.96), and the non-random taxonomic turnovers in the rest pairwise comparisons were counted as influence of dispersal limitation ($\text{RCbray} > 0.95$) or homogenizing dispersal ($\text{RCbray} < -0.95$). The rest part is called undominated. Please read the references for details.

βNTI is the standardized effect size between the observed and null values of beta mean nearest taxon distance (βMNTD). RCbray is the modified Roup-Crick metric based on taxonomic Bray-Curtis dissimilarity index.

Bigmemory (Kane et al 2013) is used to deal with large datasets.

Value

output is a list with following elements.

<code>ratio</code>	The overall percentage of turnovers governed by each process
<code>result</code>	a matrix, showing βMNTD , Bray-Curtis dissimilarity, βNTI , RC, and the identified process governing each turnover. Each row represents a turnover between two samples.
<code>pd</code>	phylogenetic distance matrix or the backingfile name if phylogenetic distance is saved by bigmemeory function.
<code>bmntd</code>	a square matrix of observed βMNTD .

BC	a square matrix of observed Bary-Curtis index.
bMNTD.rand	a matrix showing all null values of betaMNTD.
BC.rand	a matrix showing all null values of Bray-Curtis index.
setting	a data.frame showing all settings for this function.

Note

Version 6: 2020.12.5, use function `bNTI.big` to calculate `betaNTI`, which use bigmemory better.
 Version 5: 2020.9.1, remove `setwd`; change `dontrun` to `donttest` and revise `save.wd` in help doc.
 Version 4: 2020.8.21, update help document, add example. Version 3: 2016.2.15, add options to use bigmemory to handle large phylogenetic distance matrixes. Version 2: 2015.11.22

Author(s)

Daliang Ning

References

- Stegen, J.C., Lin, X., Fredrickson, J.K., Chen, X., Kennedy, D.W., Murray, C.J. et al. (2013). Quantifying community assembly processes and identifying features that impose them. *ISME J*, 7, 2069.
- Stegen, J.C., Lin, X., Fredrickson, J.K. & Konopka, A.E. (2015). Estimating and mapping ecological processes influencing microbial community assembly. *Front Microbiol*, 6, 370.
- Kane, M.J., Emerson, J., & Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[bNTIn.p](#), [bNTI.big](#), [RC.pc](#)

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree
# since pdist.big need to save output to a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

nworker=2 # parallel computing thread number
rand.time=5 # usually use 1000 for real data.

# for a small dataset, phylogenetic distance matrix can be directly used.
pd=example.data$pd
qp=qpen(comm=comm, pd=pd,
        rand.time=rand.time,nworker=nworker)

# for a big dataset, pdist.big may be used
save.wd=tempdir() # please change to the folder you want to save the pd.big output.
```



```
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)
qp2=qopen(comm=comm, pd=pd.big$pd.file, pd.big.wd=pd.big$pd.wd,
          pd.big.sname=pd.big$tip.label, tree=tree,
          rand.time=rand.time, nworker=nworker)
```

RC.bin.bigc	<i>Calculate modified Roup-Crick index based on Bray-Curtis similarity for each phylogenetic bin</i>
-------------	--

Description

Calculate modified Roup-Crick index based on Bray-Curtis similarity (RC.Bray) for each phylogenetic bin. The null model algorithm will randomize the whole community data matrix of all bins.

Usage

```
RC.bin.bigc(com, sp.bin, rand = 1000, na.zero = TRUE,
            nworker = 4, memory.G = 50, big.method = c("loop", "no"),
            weighted = TRUE, unit.sum = NULL, meta.ab = NULL,
            sig.index=c("RC", "Confidence", "SES"),
            detail.null=FALSE, output.bray=FALSE)
```

Arguments

com	community data matrix. rownames are sample names. colnames are species names.
sp.bin	one-column matrix, rownames are taxa IDs (i.e. OTU IDs), the only column shows the bin ID of each taxon. Bin IDs are integers.
rand	integer, randomization times. default is 1000.
na.zero	logic. If community data matrix has any zero-sum row (sample), Bray-Curtis index will be NA. Sometimes, this kind of NA need be set as zero to avoid some format problem in following calculation. Default is TRUE.
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
memory.G	numeric, to set the memory size as you need, so that calculation of big data will not be limited by physical memory. unit is Gb. default is 50Gb.
big.method	character, the method to handle big data. loop, randomization once after another; no, use parallel computing.
weighted	Logic, consider abundances or not (just presence/absence). default is TRUE.
unit.sum	If unit.sum is set as a number or a numeric vector, the taxa abundances will be divided by unit.sum to calculate the relative abundances, and the Bray-Curtis index in each bin will become manhattan index divided by 2. usually, unit.sum can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this special transformation.

meta.ab	a numeric vector, to define the relative abundance of each species in the regional pool. Default setting is NULL, means to calculate meta.ab as average relative abundance of each species across the samples.
sig.index	character, the index for null model significance test. RC, modified Raup-Crick index (RC) based on Bray-Curtis (BC), i.e. count the number of null BC lower than observed BC plus a half of the number of null BC equal to observed BC, to get alpha, then calculate RCbray as $(2 \times \alpha - 1)$. SES, standard effect size; Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level. default is RC. If input a vector, only the first element will be used.
detail.null	logic, if TRUE, the output will include all the null values. Default is FALSE.
output.bray	logic, if TRUE, the output will include observed Bray-Curtis.

Details

This function calculates RC.Bray in each phylogenetic bin while randomizing across all bins (Ning et al 2020). The Raup-Crick based on taxonomic dissimilarity index was proposed by Chase in 2011, and then modified to include consider species relative abundances by Stegen in 2013. The non-random part recognized by RC can reflect the influence niche selection and extreme dispersal. The original codes used a relatively time-consuming looping. This function improved the efficiency and added some parameters to fit iCAMP analysis.

SES (Kraft et al 2011) and Confidence (Ning et al 2020) are alternative significance testing indexes to evaluate how the observed beta diversity index deviates from null expectation.

Value

Output is a list.

index	list, each element is a square matrix of RC (or SES or Confidence based on Bray-Curtis) values of a bin. The elements (bins) are in the same order as in the input pdid.bin.
sig.index	character, indicates the index for null model significance test, RC, Confidence, or SES.
BC.obs	Output only if output.bray is TRUE. A list, each element is a square matrix of observed Bray-Curtis index values of a bin. The elements (bins) are in the same order as in the input pdid.bin.
rand	Output only if detail.null is TRUE. A list, each element is a matrix with null values of Bray-Curtis index for each turnover of a bin. The elements (bins) are in the same order as in the input pdid.bin.

Note

Version 5: 2020.8.19, update help document, add example. Version 4: 2020.8.2, add sig.index, detail.null, and output.bray. Version 3: 2020.6.14, add meta.ab Version 2: 2018.10.3, add unit.sum
Version 1: 2015.3.17

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.

Stegen, J.C., Lin, X., Fredrickson, J.K., Chen, X., Kennedy, D.W., Murray, C.J. et al. (2013). Quantifying community assembly processes and identifying features that impose them. *ISME J*, 7, 2069.

Kraft, N.J.B., Comita, L.S., Chase, J.M., Sanders, N.J., Swenson, N.G., Crist, T.O. et al. (2011). Disentangling the drivers of beta diversity along latitudinal and elevational gradients. *Science*, 333, 1755-1758.

See Also[RC.pc](#)**Examples**

```
data("example.data")
comm=example.data$comm
sp.bin=example.data$sp.bin
rand.time=20 # usually use 1000 for real data.
nworker=2 # parallel computing thread number
RCbin=RC.bin.bigc(com=comm, sp.bin=sp.bin,
                  rand=rand.time, nworker=nworker,
                  weighted=TRUE, sig.index="RC")
```

RC.pc

*Modified Raup-Crick index based on Bray-Curtis similarity***Description**

The Raup-Crick based on taxonomic dissimilarity index (i.e. Bray-Curtis) is to use null models to disentangle variation in community dissimilarity from variation in alpha-diversity.

Usage

```
RC.pc(comm, rand = 1000, na.zero = TRUE, nworker = 4,
      memory.G = 50, weighted = TRUE, unit.sum = NULL,
      meta.ab = NULL, sig.index=c("RC", "Confidence", "SES"),
      detail.null=FALSE, output.bray=FALSE, silent=FALSE)
```

Arguments

<code>comm</code>	Community data matrix. rownames are sample names. colnames are species names.
<code>rand</code>	integer, randomization times, default is 1000.
<code>na.zero</code>	logic. If community data matrix has any zero-sum row (sample), Bray-Curtis index will be NA. Sometimes, this kind of NA need be set as zero to avoid some format problem in following calculation. Default is TRUE.
<code>nworker</code>	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
<code>memory.G</code>	numeric, to set the memory size as you need, so that calculation of big data will not be limited by physical memory. unit is Gb. default is 50Gb.
<code>weighted</code>	Logic, whether to use abundance-weighted metrics. default is TRUE
<code>unit.sum</code>	If <code>unit.sum</code> is set as a number or a numeric vector, the taxa abundances will be divided by <code>unit.sum</code> to calculate the relative abundances, and the Bray-Curtis index in each bin will become manhattan index divided by 2. usually, <code>unit.sum</code> can be set as the sequencing depth in each sample. Default setting is NULL, means not to do this special transformation.
<code>meta.ab</code>	a numeric vector, to define the relative abundance of each species in the regional pool. Default setting is NULL, means to calculate <code>meta.ab</code> as average relative abundance of each species across the samples.
<code>sig.index</code>	character, the index for null model significance test. RC, modified Raup-Crick index (RC) based on Bray-Curtis (BC), i.e. count the number of null BC lower than observed BC plus a half of the number of null BC equal to observed BC, to get alpha, then calculate RC_{bray} as $(2 \times \alpha - 1)$. SES, standard effect size; Confidence, percentage of null values less extreme than the observed value, i.e. non-parametric one-side confidence level. default is RC. If input a vector, only the first element will be used.
<code>detail.null</code>	logic, if TRUE, the output will include all the null values. Default is FALSE.
<code>output.bray</code>	logic, if TRUE, the output will include observed Bray-Curtis.
<code>silent</code>	logic, if FALSE, some messages will show during calculation.

Details

The Raup-Crick based on taxonomic dissimilarity index was proposed by Chase in 2011, and then modified to include consider species relative abundances by Stegen in 2013. The non-random part recognized by RC can reflect the influence niche selection and extreme dispersal. The original codes used a relatively time-consuming looping. This function improved the efficiency and added some parameters to fit iCAMP analysis.

SES (Kraft et al 2011) and Confidence (Ning et al 2020) are alternative significance testing indexes to evaluate how the observed beta diversity index deviates from null expectation.

Value

Output is a list.

index	a square matrix of RC (or SES or Confidence based on Bray-Curtis) values.
sig.index	character, indicates the index for null model significance test, RC, Confidence, or SES.
BC.obs	Output only if output.bray is TRUE. A square matrix of observed Bray-Curtis index values.
rand	Output only if detail.null is TRUE. A matrix with all null values of Bray-Curtis index for each turnover.

Note

Version 6: 2020.8.19, update help document, add example. Version 5: 2020.8.2, add sig.index, detail.null, and output.bray. Version 4: 2020.6.14, add meta.ab Version 3: 2018.10.3, add unit.sum. Version 2: 2015.8.5, revise the randomization algorithm according to Stegen et al 2013. Version 1: 2015.2.12

Author(s)

Daliang Ning

References

- Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.
- Chase, J.M., Kraft, N.J.B., Smith, K.G., Vellend, M. & Inouye, B.D. (2011). Using null models to disentangle variation in community dissimilarity from variation in alpha-diversity. *Ecosphere*, 2, 1-11.
- Stegen, J.C., Lin, X., Fredrickson, J.K., Chen, X., Kennedy, D.W., Murray, C.J. et al. (2013). Quantifying community assembly processes and identifying features that impose them. *ISME J*, 7, 2069.
- Kraft, N.J.B., Comita, L.S., Chase, J.M., Sanders, N.J., Swenson, N.G., Crist, T.O. et al. (2011). Disentangling the drivers of beta diversity along latitudinal and elevational gradients. *Science*, 333, 1755-1758.

See Also

[RC.bin.bigc](#)

Examples

```
data("example.data")
comm=example.data$comm
rand.time=20 # usually use 1000 for real data.
nworker=2 # parallel computing thread number
RC=RC.pc(comm=comm, rand = rand.time,
```

```
nworker = nworker, weighted = TRUE,
sig.index="RC")
```

 snm

Estimation of neutral taxa percentae and dispersal rate

Description

To calculate the abundance-weighted or unweighted percentage of taxa following Sloan's neutral theory model. The original R code is from Burns et al (2016). Bootstrapping test and different metacommunity settings are added.

Usage

```
snm(comm, meta.com = NULL, taxon = NULL,
     alpha = 0.05, simplify = FALSE)
snm.boot(comm, rand=1000, meta.com=NULL,
         taxon=NULL, alpha=0.05, detail=TRUE)
snm.comm(comm, treat=NULL, meta.coms=NULL,
         meta.com=NULL, meta.group=NULL,
         rand=1000, taxon=NULL, alpha=0.05,
         two.tail=TRUE, output.detail=TRUE)
```

Arguments

comm	matrix or data.frame, community data, each row is a sample or site, each col-name is a taxon (a species or OTU or ASV), thus rownames should be sample IDs, colnames should be taxa IDs.
meta.com	matrix or data.frame, metacommunity data, each column represents a taxon, can be one or multiple rows. If NULL, the comm will be used to estimate relative abundance of each taxon in the metacommunity. In function snm.comm, if this is given, meta.group will be ignored.
taxon	matrix or data.frame, classification information of each taxon. Each row represents a taxon. Rownames are taxa IDs.
alpha	numeric, the significance level threshold counted as alpha value, usually 0.05.
simplify	logic, if FALSE, the function snm will perform more model fitting test and return detailed statistical information.
rand	integer, randomization times. default is 1000.
detail	logic, if TRUE, the detailed output from the function snm will be included into the output of snm.boot.
treat	matrix or data.frame, indicating the group or treatment of each sample, rownames are sample IDs. if input multiple columns, they will be analyzed one column after another.

<code>meta.coms</code>	a list, to specify the metacommunity data for each level of treatments in each of the columns of 'treat'. A basic element is a matrix, each column represents a taxon, can be one or multiple rows. If this is given, 'meta.group' and 'meta.com' will be ignored.
<code>meta.group</code>	a matrix, to specify the metacommunity ID that each sample belongs to. It should have the same column number as 'treat' if 'treat' is given. If <code>meta.coms</code> , <code>meta.com</code> , and <code>meta.group</code> are all NULL, the samples are deemed from the same metacommunity.
<code>two.tail</code>	logic, to specify the p value is calculated as two-tail (TRUE) or one-tail (FALSE).
<code>output.detail</code>	logic, if TRUE, the output of the function <code>snm.comm</code> will include all bootstrapping values.

Details

The method is developed by Burns et al (2016) based on the Sloan's model (Sloan et al 2006, 2007) which is derived from Hubbell's unified neutral theory (Hubbell 2001). According to neutral theory, the regional relative abundance and occurrence frequency of each taxon should follow a certain model (Sloan et al 2006). Thus, a taxon can be counted as 'neutral taxon' if inside a certain confidence interval of the neutral expectation, and their percentage in a sample may be used to reflect the importance of neutral processes (Burns et al 2016).

Value

Output of `snm` is a list.

<code>stats</code>	<p>output only if <code>simplify</code> is FALSE, showing statistics about model fitting and coefficients.</p> <p><code>m</code>, dispersal rate estimated by Non-linear least squares (NLS).</p> <p><code>m.ci.2.5</code> and <code>m.ci.97.5</code>, the confidence interval of <code>m</code>.</p> <p><code>m.mle</code>, dispersal rate calculated by Maximum likelihood estimation.</p> <p><code>maxLL</code>, <code>binLL</code>, and <code>poisLL</code>, maximum likelihood function value (L) for neutral theory model, binomial model, and Poisson model, respectively.</p> <p><code>Rsqr</code>, <code>Rsqr.bino</code>, and <code>Rsqr.pois</code>, R squared (coefficient of determination) of neutral theory model, binomial model, and Poisson model, respectively.</p> <p><code>RMSE</code>, <code>RMSE.bino</code>, and <code>RMSE.pois</code>, root-mean-square error.</p> <p><code>AIC</code>, <code>BIC</code>, <code>AIC.bino</code>, <code>BIC.bino</code>, <code>AIC.pois</code>, and <code>BIC.pois</code>, AIC and BIC of different models.</p> <p><code>N</code>, mean individual number in each local community.</p> <p><code>Samples</code>, sample number.</p> <p><code>Richness</code>, total number of taxa.</p> <p><code>Detect</code>, detected limitation of relative abundance in each local community, i.e. $1/N$.</p>
<code>detail</code>	<p>output only if <code>simplify</code> is FALSE. a matrix, showing detailed information of each taxon. Each row represent a taxon. Columns as blow.</p> <p><code>p</code>, observed regional relative abundance of each taxon.</p> <p><code>freq</code>, observed occurrence frequency of each taxon.</p>

freq.pred, occurrence frequency predicted by neutral theory model.
 pred.lwr and pred.upr, the confidence interval of occurrence frequency estimated by neutral theory model.
 bino.pred, bino.lwr, bino.upr, pois.pred, pois.lwr, and bino.upr, the expectation and confidence interval of occurrence frequency estimated by binomial and Poisson model, respectively.
 type, the taxon is identified as 'Neutral', or 'Below' or 'Above' the confidence interval of neutral expectation.

type.uw	the percentage (unweighted) of taxa within (Neutral), below, or above the confidence interval of the neutral theory expected frequency (given the regional relative abundance).
type.wt	the abundance weighted percentage (relative abundance sum) of taxa within (Neutral), below, or above the confidence interval of the neutral theory expectation.
sp.names	the taxa IDs for each type.

Output of `snm.boot` is a list.

<code>stats</code> , <code>detail</code> , <code>type.uw</code> , <code>type.wt</code>	output only if <code>detail</code> is TRUE. the same as output of <code>snm</code> .
<code>summary</code>	a matrix, showing observed values, mean, standard deviation, quartiles, boxplot key points and outliers, for the unweighted and weighted percentage of taxa within (Neutral), below, and above the confidence interval of neutral theory expectation.
<code>rand</code>	a matrix, showing the bootstrapping values of the unweighted and weighted percentage of different types of taxa. Each row represents one time of bootstrapping.

Output of `snm.comm` is a list.

<code>stats</code>	<code>treat.type</code> , the treatment type, a column name of the input 'treat'. <code>treatment.id</code> , the treatment name. Others are the same as the output 'stats' of <code>snm</code> . The most commonly used information is the dispersal rate values under different treatments, to investigate the effect of treatment on species dispersal.
<code>plot.detail</code>	a matrix, showing the output 'detail' of <code>snm</code> for each treatment. it is often used to show the type of each taxon and draw the figure of neutral confidence interval and each taxon.
<code>ratio.summary</code>	a matrix, showing the output 'summary' of <code>snm.boot</code> for each treatment. it is often used to draw box plots.
<code>pvalues</code>	a matrix, showing significance of difference between different treatments.
<code>boot.detail</code>	output only if <code>output.detail</code> is TRUE. A matrix, showing the output 'rand' of <code>snm.boot</code> for each treatment.

Note

Version 3: 2020.8.21, update help document, add example.

Version 2: 2018.4.16, add meta.group, meta.com, meta.coms, to consider if the samples are from different metacommunities.

Version 1: 2017.7.21

Author(s)

Daliang Ning

References

Burns, A.R., Stephens, W.Z., Stagaman, K., Wong, S., Rawls, J.F., Guillemin, K. et al. (2016). Contribution of neutral processes to the assembly of gut microbial communities in the zebrafish over host development. *ISME J*, 10, 655-664.

Sloan, W.T., Lunn, M., Woodcock, S., Head, I.M., Nee, S. & Curtis, T.P. (2006). Quantifying the roles of immigration and chance in shaping prokaryote community structure. *Environ Microbiol*, 8, 732-740.

Sloan, W.T., Woodcock, S., Lunn, M., Head, I.M. & Curtis, T.P. (2007). Modeling taxa-abundance distributions in microbial communities using environmental sequence data. *Microbial Ecology*, 53, 443-455.

Hubbell, S.P. (2001). *The unified neutral theory of biodiversity and biogeography*. Princeton University Press, Princeton, New Jersey.

Examples

```
data("example.data")
comm=example.data$comm
treat=example.data$treat
rand.time=10 # usually use 1000 for real data.
snmtest=snm.comm(comm = comm, treat = treat,
                 rand = rand.time)
```

taxa.binphy.big

Phylogenetic binning based on phylogenetic tree

Description

Phylogenetic binning for iCAMP analysis. To handle large phylogenetic tree, phylogenetic distance matrix should be calculated and saved using the package 'bigmemory' in advance.

Usage

```
taxa.binphy.big(tree, pd.desc, pd.spname, pd.wd,
               outgroup.tip = NA, outgroup.rm = TRUE,
               d.cut=NULL, ds=0.2, bin.size.limit = 24,
               nworker = 4)
```

Arguments

tree	phylogenetic tree, an object of class "phylo".
pd.desc	the name of the file to hold the backingfile description of the phylogenetic distance matrix, it is usually "pd.desc" if using default setting in pdist.big function.
pd.spname	character vector, taxa id in the same rank as the big matrix of phylogenetic distances.
pd.wd	folder path, where the bigmemory file of the phylogenetic distance matrix are saved.
outgroup.tip	a vector of tip names (i.e. OTU IDs) which is in totally different lineage from all other tips, thus can be used as outgroup to root the tree. For example, Archaeal OTUs may be set as outgroup tips when analyzing Bacterial OTUs. Default is NA, means no need to set outgroup tip.
outgroup.rm	logic, whether to remove the outgroup.tip after the tree is rooted. Default is TRUE.
d.cut	numeric, the distance from root to the truncating point of the tree.
ds	numeric, the general threshold of phylogenetic distance within which the phylogenetic signal is significant. default is 0.2.
bin.size.limit	integer, the minimal requirement of bin size (taxa numer in a bin). Default setting is 24.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.

Details

The phylogenetic tree is truncated at a certain phylogenetic distance (as short as necessary) to the root (d.cut), by which all the rest connections between tips (taxa) are lower than a threshold. Within the threshold, phylogenetic signal is generally significant. The taxa derived from the same ancestor after the truncating point are grouped to the same strict bin. Then, each small bin is merged into the bin with the nearest relatives. This procedure is repeated until all merged bins have enough taxa (\geq bin.size.limit). Bigmemory (Kane et al 2013) is used to deal with large datasets.

Value

Output is a list.

sp.bin	matrix, rownames are taxa IDs; the first column is strict bin IDs; the second column indicates which strict bin the taxon is merged into; the third column is the final bin IDs.
bin.united.sp	list, each element is a vector of taxa IDs, indicating the taxa in a final bin (after small bins are merged into nearest large bins).
bin.strict.sp	list, each element is a vector of taxa ID(s), indicating the taxa in a strict bin (before small bins are merged into large bins).

`state.strict` matrix, status of each strict bin. `bin.strict.id`, the strict bin ID; `bin.strict.taxa.num`, taxa number in each strict bin; `bin.pd.max`, `bin.pd.mean`, and `bin.pd.sd`, the maximum, mean, and standard deviation of the pairwise phylogenetic distances in each strict bin.

`state.united` matrix, status of each final bin. `bin.united.id.old`, the ID of the largest strict bin in each final bin; `bin.united.tax.num`, taxa number in each final bin; `bin.pd.max`, `bin.pd.mean`, and `bin.pd.sd`, the maximum, mean, and standard deviation of the pairwise phylogenetic distances in each strict bin.

Note

Version 3: 2020.9.1, remove `setwd`. change `dontrun` to `donttest` and revise `save.wd` in help doc.
Version 2: 2020.8.19, update help document, add example. Version 1: 2015.12.16

Author(s)

Daliang Ning

References

Ning, D., Yuan, M., Wu, L., Zhang, Y., Guo, X., Zhou, X. et al. (2020). A quantitative framework reveals ecological drivers of grassland microbial community assembly in response to warming. *Nature Communications*, 11, 4717.

Kane, M.J., Emerson, J., & Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, 55(14), 1-19. URL <http://www.jstatsoft.org/v55/i14/>.

See Also

[icamp.big](#)

Examples

```
data("example.data")
comm=example.data$comm
tree=example.data$tree

# since pd.big need to specify a certain folder,
# the following code is set as 'not test'.
# but you may test the code on your computer
# after change the folder path for 'save.wd'.

save.wd=tempdir() # please change to the folder you want to save the big niche difference matrix.
nworker=2 # parallel computing thread number
pd.big=pdist.big(tree = tree, wd=save.wd, nworker = nworker)

ds = 0.2 # setting can be changed to explore the best choice

bin.size.limit = 5 # setting can be changed to explore the best choice.
# here set as 5 just for the small example dataset.
# For real data, usually try 12 to 48.
```

```

phylobin=taxa.binphy.big(tree = tree, pd.desc = pd.big$pd.file,
                        pd.spname = pd.big$tip.label, pd.wd = pd.big$pd.wd,
                        ds = ds, bin.size.limit = bin.size.limit,
                        nworker = nworker)

```

tree.droot

Distance from root to tip(s) and node(s) on phylogenetic tree

Description

To calculate the distance from root to tip(s) and node(s) on phylogenetic tree

Usage

```
tree.droot(tree, range = NA, nworker = 4, output.path = FALSE)
```

Arguments

tree	Phylogenetic tree, an object of class "phylo".
range	NA or a vector of integer, to specify the numbering of the tips/nodes of which the distances to root will be calculated. The numbering corresponds to those in the element "edge" of the tree. Default is NA, means to calculate all tips and nodes.
nworker	integer, for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4, means 4 threads will be run.
output.path	logic, this function will call the function tree.path, if output.path is TRUE, the result of tree.path will be included in the output. Default is FALSE.

Details

A tool to get distances to root, used in phylogenetic binning.

Value

If output.path is FALSE, output is a matrix where the first column indicates the numbering of nodes/tips and the second column has the distance to root. If output.path is TRUE, output is a list with two elements.

droot	matrix, the first column indicates the numbering of nodes/tips and the second column has the distance to root.
path	result of tree.path, list of nodes and edge lengths from root to each tip and/or node.

Note

Version 2: 2020.8.19, add example. Version 1: 2015.8.19

Author(s)

Daliang Ning

See Also[tree.path](#)**Examples**

```
tree=ape::rtree(4)
nworker=2 # parallel computing thread number
droot=tree.droot(tree = tree, nworker = nworker)
```

tree.path

*List nodes and edge lengths from root to each tip and/or node***Description**

To list all the nodes and edge lengths from root to every tip and/or node.

Usage

```
tree.path(tree, nworker = 4, range = NA, cum = c("no", "from.root", "from.tip", "both"))
```

Arguments

tree	Phylogenetic tree, an object of class "phylo".
nworker	for parallel computing. Either a character vector of host names on which to run the worker copies of R, or a positive integer (in which case that number of copies is run on localhost). default is 4.
range	a numeric vector, to specify nodes and/or tips to which the path from root will be calculated. default is NA, means all tips.
cum	method to calculate cumulative branch length. "no" means not to calculate cumulative length; "from.root" means to cumulate from root to tip; "from.tip" means to cumulate from tip to root; "both" means to calculate in both ways and return both results.

Details

This function can be useful in phylogenetic diversity analysis, for example, phylogenetic distance, phylogenetic Hill number, phylogenetic binning, etc.

Value

A list result will be returned. 1st layer (the names of the list) is the end of the path, usually the names of tips and/or nodes. In 2nd layer, [[1]] is the orders of nodes between root and the tip/node specified in 1st layer; [[2]] is the edge lengths. if cum="both", [[3]] is cumulative length from root, and [[4]] is cumulative length from tip, otherwise, [[3]] is the cumulative length specified by cum.

Note

Version 1: 2016.2.14

Author(s)

Daliang Ning

See Also

[taxa.binphy.big](#), [tree.droot](#)

Examples

```
tree=ape::rtree(6)
nworker=2 # parallel computing thread number
tree.path(tree=tree, nworker=nworker, cum="both")
```

Index

- * **Other methods**
 - qpen, 78
- * **Other method**
 - snm, 86
- * **datasets**
 - example.data, 32
 - icamp.out, 45
- * **effect size**
 - cohend, 26
- * **main**
 - icamp.big, 34
 - icamp.bins, 40
 - icamp.boot, 41
 - icamp.cate, 43
 - qp.bin.js, 76
 - taxa.binphy.big, 89
- * **null model analysis**
 - NRI.p, 64
- * **package**
 - iCAMP-package, 3
- * **phylogenetic signal**
 - dniche, 30
 - ps.bin, 73
- * **phylogenetic**
 - bmntd, 4
 - bmntd.big, 6
 - bmpd, 8
 - bNRI.bin.big, 9
 - bNRIn.p, 13
 - bNTI.big, 16
 - bNTI.bin.big, 18
 - bNTIn.p, 22
 - midpoint.root.big, 60
 - mntdn, 62
 - mpdn, 63
 - NRI.p, 64
 - NTI.p, 67
 - pdist.big, 71
 - tree.droot, 92
 - tree.path, 93
- * **taxonomic**
 - RC.bin.bigc, 81
 - RC.pc, 83
- * **test**
 - change.sigindex, 25
 - null.norm, 69
- * **tool**
 - dist.3col, 28
 - dist.bin.3col, 29
 - match.2col, 56
 - match.name, 57
 - maxbigm, 59
- bmntd, 4, 21, 24
- bmntd.big, 6, 18
- bmpd, 8, 12, 15
- bNRI.bin.big, 9
- bNRIn.p, 9, 12, 13
- bNTI.big, 16, 80
- bNTI.bin.big, 18
- bNTIn.p, 6, 21, 22, 80
- change.sigindex, 25, 70
- cohend, 26
- dist.3col, 28
- dist.bin.3col, 29
- dniche, 30, 74, 75
- example.data, 32
- iCAMP (iCAMP-package), 3
- iCAMP-package, 3
- icamp.big, 26, 34, 40–43, 70, 77, 91
- icamp.bins, 40, 44
- icamp.boot, 41
- icamp.cate, 43
- icamp.out, 45
- match.2col, 56

match.name, [57](#)
maxbigm, [59](#), [61](#)
midpoint.root.big, [60](#), [60](#)
mntdn, [62](#), [68](#)
mpdn, [63](#), [66](#)

NRI.p, [64](#), [64](#)
NTI.p, [63](#), [67](#)
null.norm, [26](#), [69](#)

pdist.big, [71](#)
ps.bin, [32](#), [73](#)

qp.bin.js, [39](#), [76](#)
qpen, [18](#), [78](#)

RC.bin.bigc, [81](#), [85](#)
RC.pc, [80](#), [83](#), [83](#)

snm, [86](#)

taxa.binphy.big, [73](#), [75](#), [89](#), [94](#)
tree.droot, [92](#), [94](#)
tree.path, [93](#), [93](#)