

Package: comstab (via r-universe)

August 22, 2024

Title Partitioning the Drivers of Stability of Ecological Communities

Version 0.0.3.9000

Description Contains the basic functions to apply the unified framework for partitioning the drivers of stability of ecological communities. Segrestin et al. (2024) <doi:10.1111/geb.13828>.

License GPL-3

URL <https://github.com/jsegrestin/comstab>

BugReports <https://github.com/jsegrestin/comstab/issues>

Imports graphics, stats, Ternary, smatr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository <https://jsegrestin.r-universe.dev>

RemoteUrl <https://github.com/jsegrestin/comstab>

RemoteRef HEAD

RemoteSha 250ced902f030542aa45efc5399955a9b86723f3

Contents

comTS	2
partitionR	3
plot.comstab	5
plot.TPL	6
ternStab	7
TPL	8

Index	9
--------------	----------

comTS

*Simulating time series for ecological communities***Description**

comTS() is a function used to simulate community Time Series based on custom parameters.

Usage

```
comTS(nsp, ny, even, mvs, sync = c("0", "1", "2", "-1", "-2"))
```

Arguments

nsp	Number of species in the community (integer).
ny	Length of the time series in years (integer > 5).
even	Parameter of the geometric rank-abundance curve ranging between 0 and 1 (numeric).
mvs	Scaling coefficient of the mean-variance relationship ranging between 1 and 2 (numeric).
sync	Level of synchrony between species (factor): <ul style="list-style-type: none"> • "0" (defaults): independant fluctuations. • "1": positive synchrony. • "2": high positive synchrony. • "-1": anti-synchronous fluctuations. • "-2": high anti-synchronous fluctuations.

Details

The simulation produces temporal series of abundances of 'nsp' species for 'ny' years. The mean abundance of each species is determined by a geometric series of 'nsp' values using a constant 'even' ratio between successive terms. Hence, a high 'even' value means a community with even species abundances while a low 'even' means a strongly dominated community. Species temporal variances are calculated following Taylor's power law, using with a 'mvs' scaling coefficient. Finally, for each species, the 'ny' abundance values are sampled from a normal distribution with the corresponding species parameters, and bounded to positive values. Since each species is simulated independently of others, the resulting simulation can be used to describe a community where the 'nsp' species fluctuate independently ('sync' = "0"). To simulate an overall positive synchrony ('sync' = "1"), the temporal sequences of each species are sorted to maximize the number of years with all species having values above or below their respective median (one random selection among many possible combinations). A stabilizing negative synchrony ('sync' = "-1") is simulated by sorting the temporal sequences of each species to maximize the number of years where successively abundant species have values above and below their respective median (one random selection among many possible combinations). High positive synchrony ('sync' = "2") and high negative synchrony ('sync' = "-2") are generated using a similar approach but sorting values based on four quartiles instead of using the median only.

The simulation uses a simplistic approach where species fluctuations are not related to any underlying environmental factor nor demographic parameters. Consequently, the temporal order of the simulated abundances for each species cannot be considered realistic. Nevertheless, this simplification has little influence on the analyses performed in this R package since none of the indices calculated depend on the temporal order of individual series, but rather describe the overall variability and temporal coordination of species.

Value

A matrix of 'ny' rows and 'nsp' columns, containing numerical values of species abundances. The parameters used to compute species values (even, mvs, and sync) are stored as attributes of the matrix.

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

Examples

```
require(stats)

comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")
```

partitionR

Partitioning of the temporal CV of ecological communities

Description

PartitionR() is a function used to partition the temporal coefficient of variation of a community into the variability of the average species and three stabilizing effects: the dominance, asynchrony and averaging effects (see Details).

Usage

```
partitionR(z, ny = 0)
```

Arguments

z	A matrix containing repeated measurements of species abundances. The matrix must contain numerical values only, with years in rows and species in columns. Remove any extra column.
ny	Only species appearing more than ny years (integer, defaults to 0) are used in the calculations.

Details

The analytic framework is described in details in Segrestin *et al.* (2024). In short, the partitioning relies on the following equation:

$$CV_{com} = CV_e \Delta \Psi \omega$$

where CV_{com} is the community coefficient of variation (reciprocal of community stability), CV_e is the expected community CV when controlling for the dominance structure and species temporal synchrony, Δ is the dominance effect, Ψ is the asynchrony effect, and ω is the averaging effect.

Value

Returns an object of class 'comstab'.

An object of class 'comstab' is a list containing the following components:

- 'CVs' a named vector of calculated coefficient of variations. CV_e is the CV of an average species, $CV_{\tilde{}}$ is the mean of species CVs weighted by their relative abundances, CV_a is the expected community CV if the community was stabilized by species asynchrony only, and CV_c is the observed community CV.
- 'Stabilization' a named vector of the stabilizing effects. τ is the total stabilization, Δ is the dominance effect, Ψ is the asynchrony effect, and ω is the averaging effect.
- 'Relative' a named vector of the relative contributions of each stabilizing effect to the total stabilization. Δ_{cont} , Ψ_{cont} , and ω_{cont} are the relative contribution of respectively, the dominance, asynchrony, and averaging effects to the total stabilization. Returns a vector of NAs if any Stabilizing effect is higher than 1.

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

References

Segrestin *et al.* (2024) A unified framework for partitioning the drivers of stability of ecological communities. *Global Ecology and Biogeography*, 33(5), e13828. <https://doi.org/10.1111/geb.13828>

Examples

```
require(stats)

# Simulates a custom community time series using 'comTS()':
z <- comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")

# Runs the partitioning of the community coefficient of variation:
partitionR(z)
```

`plot.comstab`*Plotting a comstab object*

Description

Plotting method for object inheriting from class "comstab".

Usage

```
## S3 method for class 'comstab'  
plot(x, ..., xlab = "", ylab = "Log scale", cex.comp = 1)
```

Arguments

<code>x</code>	object of class 'comstab'.
<code>...</code>	other parameters to be passed through to plotting functions.
<code>xlab</code>	a label for the x axis, removed by defaults.
<code>ylab</code>	a label for the y axis, defaults to 'Log scale'.
<code>cex.comp</code>	A numerical value giving the label size of stabilizing components. This is an absolute measure, not scaled by <code>par("cex")</code> .

Value

No return value, graphical function.

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

Examples

```
require(graphics)  
  
# Simulates a custom community time series using 'comTS()':  
z <- comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")  
  
# Runs the partitioning of the community coefficient of variation:  
x <- partitionR(z)  
  
# Plots the result  
plot(x)
```

`plot.TPL`*Plotting a TPL object*

Description

Plotting method for object inheriting from class "TPL".

Usage

```
## S3 method for class 'TPL'  
plot(x, ..., xlab = "Mean", ylab = "Variance", log = "xy")
```

Arguments

<code>x</code>	object of class 'comstab'.
<code>...</code>	other parameters to be passed through to plotting functions.
<code>xlab</code>	a label for the x axis, defaults to 'Mean'.
<code>ylab</code>	a label for the y axis, defaults to 'Variance'.
<code>log</code>	Defaults to 'xy' to show the log-log relationship.

Value

No return value, graphical function.

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

Examples

```
require(graphics)  
  
# Simulates a custom community time series using 'comTS()':  
z <- comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")  
  
# Computes Taylor's power law:  
x <- TPL(z)  
  
# Plots the result  
plot(x)
```

ternStab	<i>Plotting the relative contribution of stabilizing effects</i>
----------	--

Description

ternStab() is a graph function used to represent the relative contributions of the three stabilizing effects ("Dominance", "Asynchrony" and "Averaging") on a ternary plot.

Usage

```
ternStab(x, ..., point = TRUE, add = FALSE)
```

Arguments

x	object of class 'comstab'.
...	other parameters to be passed through to plotting functions.
point	plot the community on the ternary plot (logical, defaults to TRUE)
add	add the community on the current plot window (logical, defaults to FALSE)

Value

No return value, graphical function.

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

Examples

```
require(Ternary)

# Simulates a custom community time series using 'comTS()':
z <- comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")

# Runs the partitioning of the community coefficient of variation:
x <- partitionR(z)

# Plots the relative contributions
oldpar <- par(no.readonly = TRUE)
par(mar = c(0, 0, 0, 0))
ternStab(x)

# Adds a second community on the ternary plot
z2 <- comTS(nsp = 15, ny = 30, even = .7, mvs = 1.1, sync = "1")
x2 <- partitionR(z2)
ternStab(x2, add = TRUE, col = "red")

par(oldpar)
```

 TPL

Computing Taylor's power law

Description

TPL() is a function used to compute the power function between species temporal means and variances.

Usage

```
TPL(z, ny = 0, method = c("OLS", "SMA"))
```

Arguments

- | | |
|--------|--|
| z | A matrix containing repeated measurements of species abundances. The matrix must contain numerical values only, with years in rows and species in columns. Remove any extra column. |
| ny | Only species appearing more than ny years (integer, defaults to 0) are used in the calculations. |
| method | Regression method used to estimate the scaling coefficient (factor): <ul style="list-style-type: none"> • OLS (defaults) : Ordinary least square • SMA: Standardized major axis, sometimes called reduced major axis |

Value

An object of class 'TPL' is a list containing the following components:

- data a data.frame containing species temporal means and variance
- test the result of the Pearson's correlation test between log10-transformed species means and variances (see stats::cor.test)
- a the intercept of the power law
- b the scaling coefficient of the power law

Author(s)

Jules Segrestin, <jsegrestin@gmail.com>

Examples

```
require(smatr)

# Simulates a custom community time series using 'comTS()':
z <- comTS(nsp = 10, ny = 30, even = 0.6, mvs = 1.5, sync = "0")

# Computes Taylor's power law:
TPL(z)
```


Index

comTS, [2](#)

partitionR, [3](#)

plot.comstab, [5](#)

plot.TPL, [6](#)

ternStab, [7](#)

TPL, [8](#)