

Package ‘betaStability’

October 9, 2026

Title Quantify the Compositional Stability of Each Community Based on a Single Sampling Event

Version 0.1.0

Description Quantify the stability of each community based on the beta diversity between communities gathered in a single sampling event rather than a series of continuous sampling activities.

URL <https://github.com/gaoyu19920914/betaStability/>

BugReports <https://github.com/gaoyu19920914/betaStability/issues/>

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr, rmarkdown

Imports BBmisc, elevatr, gdm, ggplot2, glmnet, mgcv, randomForest, raster, reshape2, stats, usedist, vegan, xgboost

Suggests BiocManager, knitr, rioja, rmarkdown, sf, testthat (>= 3.0.0)

biocViews StatisticalMethod, Software

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Yu Gao [aut, cre, cph, fnd] (ORCID:
<https://orcid.org/0000-0003-4406-2978>)

Maintainer Yu Gao <gaoyu19920914@gmail.com>

Repository CRAN

Date/Publication 2026-10-09 03:50:02 UTC

Contents

betaStability	2
calcStability	4
gamPred	5
gdmPred	6
glmPred	7
linearPred	8
mlPred	9
plotCorrelation	9
plotStability	10
rfPred	11
varecoords	12
xgboostPred	13
Index	14

betaStability	<i>Beta Stability Calculation with Multiple Prediction Methods</i>
---------------	--

Description

This function integrates various prediction methods (linear, multiple linear, generalized linear model, generalized additive model, generalized dissimilarity model, random forest, and xgboost) to calculate the stability of each site.

Usage

```
betaStability(  
  comtable = NULL,  
  envmeta = NULL,  
  comdist = NULL,  
  envdist = NULL,  
  sitenames = NULL,  
  symmetric = FALSE,  
  method = "linearPred",  
  X = NULL,  
  Y = NULL,  
  geo_enabled = TRUE,  
  GAM.dist.method = "manhattan",  
  vegdist.method = "bray",  
  xgboost.params = NULL,  
  seed = 42  
)
```

Arguments

<code>comtable</code>	The community table (required).
<code>envmeta</code>	The environmental metadata table/matrix (required).
<code>comdist</code>	The community dissimilarity matrix (optional). If not provided, computed from <code>comtable</code> using <code>vegdist()</code> .
<code>envdist</code>	The environmental dissimilarity matrix (optional). If not provided, computed from <code>envmeta</code> using euclidean distance on range-normalized <code>envmeta</code> .
<code>sitenames</code>	The names of the site (optional, default: <code>NULL</code>)
<code>symmetric</code>	Whether to use symmetric algorithm in the calculating (optional, default: <code>FALSE</code>)
<code>method</code>	A character string or vector specifying the prediction method(s) to use. Available options: "linearPred", "mlPred", "glmPred", "gamPred", "gdmPred", "rfPred", "xgboostPred". Use "all" to run all methods. Default is "linearPred".
<code>X</code>	The X coordinates of the sites for <code>gdmPred</code> (optional, default: <code>NULL</code>)
<code>Y</code>	The Y coordinates of the sites for <code>gdmPred</code> (optional, default: <code>NULL</code>)
<code>geo_enabled</code>	Whether to include geographic info for <code>gdmPred</code> (default: <code>TRUE</code>)
<code>GAM.dist.method</code>	The method for calculating dist for <code>gamPred</code> (default: "manhattan")
<code>vegdist.method</code>	The method for calculating <code>comdist</code> from <code>comtable</code> (default: "bray")
<code>xgboost.params</code>	A list of parameters for the xgboost model (default: <code>NULL</code>). If <code>NULL</code> , default parameters will be used.
<code>seed</code>	The random seed for reproducibility of <code>rfPred</code> and <code>xgboostPred</code> (default: 42)

Value

If `method = "all"`, returns a data frame with 7 columns, each representing results from each selected method. If `method` length is 1, returns a column vector of predicted stability values. If `method` length > 1, returns a data frame with each column representing results from each selected method.

Examples

```
library(vegan)
data(varespec)
data(varechem)

# Single method (linearPred)
result_linear <- betaStability(
  comtable = varespec,
  envmeta = varechem,
  method = "linearPred"
)

# Multiple methods
results_multi <- betaStability(
  comtable = varespec,
  envmeta = varechem,
  method = c("linearPred", "mlPred")
)
```

)

calcStability*calculate stability based on predicted and measured distances.*

Description

This function calculates the stability of a site by comparing the predicted distance and the measured distance. The algorithm is simple and straightforward: if the measured distance is greater than the predicted distance, the stability is negative, and if the measured distance is less than the predicted distance, the stability is positive. The stability value is normalized to be between -1 and 1, where -1 indicates the least stable (measured distance is much greater than predicted) and 1 indicates the most stable (measured distance is much less than predicted). To avoid magnifying the deviations near 0 and 1 of the predicted.dist, a symmetric algorithm is introduced without changing the range of the calculated stability value (-1 to 1).

Usage

```
calcStability(predicted.dist, measured.dist, symmetric = FALSE)
```

Arguments

`predicted.dist` The predicted distance (shall be in range 0~1)
`measured.dist` The measured distance (shall be in range 0~1)
`symmetric` Whether to use symmetric algorithm in the calculating (optional, default: FALSE).

Value

a numeric value of stability for the site in range [-1, 1].

Examples

```
calcStability(predicted.dist = 0.3, measured.dist = 0.5)  
calcStability(predicted.dist = 0.3, measured.dist = 0.1)  
calcStability(predicted.dist = 0.3, measured.dist = 0.5, symmetric = TRUE)  
calcStability(predicted.dist = 0.3, measured.dist = 0.1, symmetric = TRUE)
```

gamPred	<i>calculation of stability using an generalized additive model.</i>
---------	--

Description

This function will take the community count table and the environmental metadata table as input, and calculate the stability of each site using a generalized additive model (GAM). Alternatively, if no dissimilarity matrix of the community is provided, the function will calculate the community dissimilarity based on Bray-Curtis distance and use it. In GAM, the prediction results are expected to perform better than the linear models.

Usage

```
gamPred(  
  comtable,  
  envmeta,  
  comdist = NULL,  
  sitenames = NULL,  
  GAM.dist.method = "manhattan",  
  symmetric = FALSE  
)
```

Arguments

comtable	The community table
envmeta	The environmental metadata table/matrix
comdist	The community dissimilarity matrix (optional, default: NULL)
sitenames	The names of the site (optional, default: NULL)
GAM.dist.method	The method for calculating dist (default: "manhattan")
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)  
data(varespec)  
data(varechem)  
example.stability_GAM <- gamPred(varespec, varechem)
```

gdmPred

*calculation of stability using a generalized dissimilarity model.***Description**

This function will take the community dissimilarity matrix and the environmental metadata table/matrix as input, and make predictions based on a generalized dissimilarity model (GDM), with optional geographic information (X and Y can be longitude and latitude). This model considers the nonlinear relationship between community dissimilarity and environmental distance, and can also include geographic distance as a predictor.

Usage

```
gdmPred(
  comdist,
  envmeta,
  sitenames = NULL,
  X = NULL,
  Y = NULL,
  geo_enabled = TRUE,
  symmetric = FALSE
)
```

Arguments

comdist	The community dissimilarity matrix
envmeta	The environmental metadata table/matrix
sitenames	The names of the site (optional, default: NULL)
X	The X coordinates of the sites (optional, default: NULL)
Y	The Y coordinates of the sites (optional, default: NULL)
geo_enabled	Whether to include geographic info (default: TRUE)
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)
data(varespec)
data(varechem)
data(BCI)
data(BCI.env)
example.comdist <- vegdist(varespec)
example.stability_GDM <- gdmPred(example.comdist, varechem)
```

```

example.stability_GDM_geo <- gdmPred(vegdist(BCI, "bray"),
  BCI.env[, c("Precipitation", "Elevation", "EnvHet")],
  X = BCI.env$UTM.EW,
  Y = BCI.env$UTM.NS
)

```

glmPred

calculation of stability using a generalized linear model.

Description

This function will take the dissimilarity matrix and the environmental matrix as input, and calculate the stability of each site using a generalized linear model (gLm), where the contributions are constrained as non-negative lower.limits=0 to ensure the explainability of each coefficient. The stability is calculated by comparing the predicted distance (based on the linear model) and the mean measured distance (based on vegdist function).

Usage

```
glmPred(comdist, envmeta, sitenames = NULL, symmetric = FALSE)
```

Arguments

comdist	The community dissimilarity matrix
envmeta	The environmental metadata table/matrix
sitenames	The names of the site
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```

library(vegan)
data(varespec)
data(varechem)
example.comdist <- vegdist(varespec)
example.stability_GLM <- glmPred(example.comdist, varechem)

```

linearPred	<i>calculation of stability using linear prediction model.</i>
------------	--

Description

This function will take the diversity matrix and the environmental distance matrix as input, and calculate the stability of each site using linear model. The stability is calculated by comparing the predicted distance (based on the linear model) and the mean measured distance (based on vegdist function).

Usage

```
linearPred(comdist, envdist, sitenames = NULL, symmetric = FALSE)
```

Arguments

comdist	The community dissimilarity matrix
envdist	The environmental dissimilarity matrix
sitenames	The names of the site
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)
library(stats)
data(varespec)
data(varechem)
example.comdist <- vegdist(varespec)
example.envdist <- dist(
  BBmisc::normalize(
    varechem,
    method = "range",
    margin = 2
  ),
  method = "euclidean"
)
example.stability_LM <- linearPred(example.comdist, example.envdist)
```

mlPred	<i>calculation of stability using multiple linear regression model.</i>
--------	---

Description

This function will take the diversity matrix and the environmental distance matrix as input, and calculate the stability of each site using multiple linear model (ML). The stability is calculated by comparing the predicted distance (based on the multiple linear model) and the mean measured distance between the site and other sites (based on the difference of envmeta and the corresponding comdist).

Usage

```
mlPred(comdist, envmeta, sitenames = NULL, symmetric = FALSE)
```

Arguments

comdist	The community dissimilarity matrix
envmeta	The environmental metadata table/matrix
sitenames	The names of the site
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)
data(varespec)
data(varechem)
example.comdist <- vegdist(varespec)
example.stability_ML <- mlPred(example.comdist, varechem)
```

plotCorrelation	<i>Plot Correlation Matrix</i>
-----------------	--------------------------------

Description

This function takes a result dataframe from betaStability() and creates a faceted scatter plot matrix to visualize correlations between different stability quantification methods.

Usage

```
plotCorrelation(data, method = "spearman")
```

Arguments

data	A dataframe containing stability results from betaStability(). Must have at least 2 numeric columns.
method	Correlation method to use. Default is "spearman". Other options include "pearson" and "kendall".

Value

A ggplot2 plot object showing pairwise correlations between columns.

Examples

```
library(vegan)
library(ggplot2)
data(varespec)
data(varechem)
results <- betaStability(
  comtable = varespec,
  envmeta = varechem,
  method = c("linearPred", "mlPred", "glmPred")
)
plotCorrelation(results)
```

plotStability

Plot Stability Results

Description

Creates a point plot of the stability results produced by betaStability(). If a coords argument is supplied, a map of the sites is drawn instead (optionally with an elevation raster as background).

Usage

```
plotStability(stability_result, coords = NULL, sitenames = NULL, elev = FALSE)
```

```
plotStabilityMap(stability_result, coords, sitenames = NULL, elev = FALSE)
```

Arguments

stability_result	The output from betaStability().
coords	Optional data frame of site coordinates. Columns may be named x/y, X/Y, lon/lat, or longitude/latitude; they are normalised internally. When supplied, plotStability() behaves like plotStabilityMap().
sitenames	Optional vector of site names. If not provided, uses rownames from stability_result. Users shall make sure the provided sitenames correspond to the rownames of stability_result in the correct order.

elev Logical. Only used when **coords** is supplied. If **TRUE**, an elevation raster is drawn behind the points using **elevatr**.

Value

A **ggplot2** plot object.

Examples

```
library(vegan)
library(ggplot2)
data(varespec)
data(varechem)
data(varecoords)
results <- betaStability(
  comtable = varespec,
  envmeta = varechem, method = c("linearPred", "mlPred")
)
plotStability(results)
plotStability(results, coords = varecoords)
## Not run:
plotStability(results, coords = varecoords, elev = TRUE)

## End(Not run)

# Alias
plotStabilityMap(results, varecoords)
```

rfPred	<i>calculation of stability using a random forest model.</i>
--------	--

Description

This function will take the dissimilarity matrix and the environmental matrix as input, and calculate the stability of each site using a random forest model to improve the prediction performance.

Usage

```
rfPred(comdist, envmeta, sitenames = NULL, seed = NULL, symmetric = FALSE)
```

Arguments

comdist	The community dissimilarity matrix
envmeta	The environmental metadata table/matrix
sitenames	The names of the site
seed	The random seed for reproducibility of the random forest model
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)
data(varespec)
data(varechem)
example.comdist <- vegdist(varespec[1:10,])
example.stability_RF <- rfPred(example.comdist, varechem[1:10,])
```

varecoords

Coordinates data

Description

A dataset of coordinate information for `vegan::varechem` retrieved by georeferencing the Fig.1 of the original article "Effects of reindeer grazing on understorey vegetation in dry *Pinus sylvestris* forests" by Väre et al. (1995). NOTICE: The data is not precise nor original.

Usage

```
varecoords
```

Format

varecoords:

A data frame with 28 rows and 3 columns:

site site name/number

x longitude

y latitude ...

Source

<https://onlinelibrary.wiley.com/doi/10.2307/3236351>

xgboostPred	<i>calculation of stability using a xgboost model.</i>
-------------	--

Description

This function will take the dissimilarity matrix and the environmental matrix as input, and calculate the stability of each site using a xgboost model to improve the prediction performance.

Usage

```
xgboostPred(  
  comdist,  
  envmeta,  
  sitenames = NULL,  
  seed = NULL,  
  params = NULL,  
  symmetric = FALSE  
)
```

Arguments

comdist	The community dissimilarity matrix
envmeta	The environmental metadata table/matrix
sitenames	The names of the site
seed	The random seed for reproducibility of the xgboost model
params	A list of parameters for the xgboost model. If NULL, default parameters will be used.
symmetric	Whether to use symmetric algorithm in the calculating.

Value

a column vector of predicted stability values for each site

Examples

```
library(vegan)  
data(varespec)  
data(varechem)  
example.comdist <- vegdist(varespec[1:10,])  
example.stability_XGB <- xgboostPred(example.comdist, varechem[1:10,])
```

Index

* **datasets**

varecoords, [12](#)

betaStability, [2](#)

calcStability, [4](#)

gamPred, [5](#)

gdmPred, [6](#)

glmPred, [7](#)

linearPred, [8](#)

m1Pred, [9](#)

plotCorrelation, [9](#)

plotStability, [10](#)

plotStabilityMap (plotStability), [10](#)

rfPred, [11](#)

varecoords, [12](#)

xgboostPred, [13](#)