

Package ‘STEPCAM’

May 6, 2025

Type Package

Title ABC-SMC Inference of STEPCAM

Version 1.2.3

Description Collection of model estimation, and model plotting functions related to the STEPCAM family of community assembly models. STEPCAM is a STEPwise Community Assembly Model that infers the relative contribution of Dispersal Assembly, Habitat Filtering and Limiting Similarity from a dataset consisting of the combination of trait and abundance data. See also <[doi:10.1890/14-0454.1](https://doi.org/10.1890/14-0454.1)> for more information.

License GPL-2

URL <https://github.com/thijsjanzen/STEPCAM>

BugReports <https://github.com/thijsjanzen/STEPCAM/issues>

Imports ade4, ape, FD, geometry, grid, gtools, stats

Suggests devtools, knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Thijs Janzen [aut, cre],
Fons van der Plas [aut],
Thorsten Hauffe [ctb]

Maintainer Thijs Janzen <thijsjanzen@gmail.com>

Repository CRAN

Date/Publication 2025-05-06 14:00:07 UTC

Contents

STEPCAM-package	2
generate.Artificial.Data	3
plotSMC	4
plotSTEPCAM	5
STEPCAM_ABC	6
TernPlot	8

STEPCAM-package *ABC-SMC Inference of STEPCAM*

Description

ABC-SMC inference of the relative contribution of dispersal assembly, filtering and limiting similarity through the use of stepwise community assembly.

Updates

Version 1.2

- STEPCAM can now be found on GitHub: <https://github.com/thijsjanzen/STEPSCAM>
- Merged STEPCAM code with the supplementary code of Hauffe et al. 2016
- Added two vignettes
- Added a large number of code tests

Version 1.1.2

- updated the prior - now all three processes have a mean prior value of 1/3, and a log-normal kind shape in [0,1] Version 1.1.1
- fixed a bug in the ABC-SMC code leading to the error "Error in if (fit < threshold) : missing value where TRUE/FALSE needed"

Version 1.1

- calculation of the diversity indices is split up into two parts: calculating the ordination axes, and the actual calculation of the indices - this allows for a speed up of the code by moving the calculation of the ordination axes outside the SMC algorithm. Many thanks to Torsten Hauffe for this improvement!
- Corrected scaling of the dbFD function - it now always treats abundances as presence/absence data, much thanks to Torsten Hauffe for pointing this out!
- updated references

Details

Package: STEPCAM
 Type: Package
 Version: 1.2
 Date: 2016-09-20
 License: GPL 2.0

The function STEPCAM_ABC performs an ABC-SMC analysis of the STEPCAM model on user provided data, the functions plotSMC, plotSTEPSCAM and TernPlot provide functions to plot the output generated by the ABC-SMC analysis.

Author(s)

Thijs Janzen & Fons van der Plas

Maintainer: Thijs Janzen <thijsjanzen@gmail.com>

References

Fons van der Plas, Thijs Janzen, Alejandro Ordonez, Wimke Fokkema, Josephine Reinders, Rampal S. Etienne, and Han Olf 2015. A new modeling approach estimates the relative importance of different community assembly processes. *Ecology* 96:1502-1515.

Hauffe, Torsten, Christian Albrecht, and Thomas Wilke. Assembly processes of gastropod community change with horizontal and vertical zonation in ancient Lake Ohrid: a metacommunity speciation perspective. *Biogeosciences* 13.10 (2016): 2901-2911.

generate.Artificial.Data

Generate Artificial data that can be used by the STEPCAM model

Description

This function generates artificial data with a given number of species, traits and communities. The generated data can directly be used in STEPCAM-ABC inference.

Usage

```
generate.Artificial.Data(n_species, n_traits, n_communities,
occurence_distribution, average_richness, sd_richness, mechanism_random)
```

Arguments

n_species	The number of species in the species pool (so across all communities) of the desired dataset.
n_traits	The number of traits in the desired dataset.
n_communities	The number of communities in the desired dataset.
occurence_distribution	species occurrence distribution (analogous to species abundance distribution) in species pool. A lognormal distribution is assumed. The higher the value, the more uneven the occurrence distribution in the species pool
average_richness	average richness in local community as a proportion of richness in species pool
sd_richness	standard deviation in richness across communities
mechanism_random	if TRUE, species are randomly drawn from from species pool. If FALSE, it is assumed that a certain trait combination is optimal in each community and species with traits similar to this optimum more likely end up in the local community. Therefore, random.mechanism = FALSE assumes filtering processes to be most dominant in community assembly

Value

traits	A matrix containing the traits for all species, rows are the different species, column correspond to the respective traits.
abundances	A matrix containing the abundances of all species for all plots. Rows correspond to species, columns to plots.

Author(s)

Fons van der Plas

Examples

```
Artificial.Data <- generate.Artificial.Data(n_species=40, n_traits=3, n_communities = 5,
  occurrence_distribution = 0.2, average_richness = 0.5, sd_richness = 0.2,
  mechanism_random = FALSE)
```

plotSMC

Plot the progression of the SMC algorithm.

Description

This function plots the progression over the several iterations of the ABC-SMC algorithm. It can be used to explore results obtained with the STEPCAM-ABC function.

Usage

```
plotSMC(path)
```

Arguments

path Working Directory

Author(s)

Thijs Janzen

Examples

```
## Not run:
Artificial.Data <- generate.Artificial.Data(n_species = 40, n_traits = 3,
  n_communities = 5, occurrence_distribution = 0.2,
  average_richness = 0.5, sd_richness = 0.2,
  mechanism_random=FALSE)
O <- STEPCAM_ABC(Artificial.Data$abundances, Artificial.Data$traits,
  numParticles = 10, n_traits = 3, plot_number = 1, stopRate = 0.8)
currentDir <- getwd();
plotSMC(paste(currentDir, "/", sep=""));

## End(Not run)
```

`plotSTEPCAM`*Plot the output generated by the STEP CAM_ABC function*

Description

This function plots the posterior distribution inferred by the STEP CAM_ABC function for the three community assembly parameters dispersal assembly, habitat filtering and limiting similarity.

Usage`plotSTEPCAM(output)`**Arguments**

`output` The list generated by the function STEP CAM_ABC as output, this list contains the following elements:

-
- DA a vector containing the posterior distribution for the parameter dispersal assembly
-
- HF a vector containing the posterior distribution for the parameter habitat filtering
-
- LS a vector containing the posterior distribution for the parameter limiting similarity

Author(s)

Thijs Janzen

Examples

```
## Not run:
Artificial.Data <- generate.Artificial.Data(n_species = 40, n_traits = 3,
                                           n_communities = 5, occurrence_distribution = 0.2,
                                           average_richness = 0.5, sd_richness = 0.2,
                                           mechanism_random=FALSE)
O <- STEP CAM_ABC(Artificial.Data$abundances,Artificial.Data$traits,
                  numParticles=10,n_traits=3,plot_number=1,stopRate=0.8);
plotSTEPCAM(O);

## End(Not run)
```

STEPCAM_ABC

*ABC-SMC inference of the STEPCAM model***Description**

ABC-SMC inference of the relative contribution of stochasticity (or random removal), filtering and limiting similarity (e.g. of shared resources (competition) or predators) through the use of stepwise community assembly and fitting Functional Diversity patterns of simulated communities with those of observed communities. It takes quite a lot of computing time to run this function, so it is recommended to first run it for a very simple (species poor, artificial) dataset (see example).

Usage

```
STEPCAM_ABC(data_abundances, data_species, numParticles, n_traits,
            plot_number, stopRate,
            stop_at_iteration,
            continue_from_file)
```

Arguments

data_abundances

- data_abundances is a matrix containing the abundances of all species over all plots.
- rows every row corresponds to a different plot or community
- columns every column corresponds to a different species.
For an example of a suitable data_abundances matrix, run the generate.Artificial.Data function

data_species

- data_species is a matrix containing the species' specific traits.
- rows every row corresponds to a different species
- columns the first column contains the name of the species, every consecutive column corresponds to a different trait
It is important to realize that ideally, the number of traits is lower than the number of species in the most species poor plot of the 'data_abundances' matrix, as FRic calculation requires at least as many species as their are traits, and will automatically omit some traits if this assumption is not met.
For an example of a suitable data_species matrix, run the generate.Artificial.Data function

numParticles

The number of particles to be used in the ABC-SMC analysis.

n_traits

The number of traits used in the analysis.

plot_number

Row number of the plot for which we want to infer the contributions of stochasticity, filtering and competition.

stopRate	The SMC procedure stops as soon as the acceptance rate of newly proposed parameter combinations drops below this level. Beware that low values for this parameter might lead into long computation times.
stop_at_iteration	If the user instead wants to perform a fixed number of iterations, this parameter sets the maximum number of iterations (the SMC stops when either the maximum number of iterations is reached, or the acceptance rate drops below the stopRate, whatever happens first)
continue_from_file	Should the ABC-SMC algorithm continue from previous output of the STEPCAM_ABC function?

Value

DA	Posterior distribution of the relative contribution of dispersal assembly (random removal steps) to community assembly
HF	Posterior distribution of the relative contribution of habitat filtering to community assembly
LS	Posterior distribution of the relative contribution of limiting similarity to community assembly

Author(s)

Thijs Janzen & Fons van der Plas + Torsten Hauffe

References

Fons van der Plas, Thijs Janzen, Alejandro Ordonez, Wimke Fokkema, Josephine Reinders, Rampal S. Etienne, and Han Olf 2015. A new modeling approach estimates the relative importance of different community assembly processes. *Ecology* 96:1502-1515.

Examples

```
## Not run:
Artificial.Data <- generate.Artificial.Data(n_species = 40, n_traits = 3,
                                           n_communities = 5, occurrence_distribution = 0.2,
                                           average_richness = 0.5, sd_richness = 0.2,
                                           mechanism_random=FALSE)
O <- STEPCAM_ABC(Artificial.Data$abundances, Artificial.Data$traits,
                 numParticles = 10, n_traits = 3, plot_number = 1, stopRate = 0.8)

## End(Not run)
```

TernPlot

Create a ternary plot from results of the STEPCAM_ABC function

Description

This function plots the obtained combinations of dispersal assembly, habitat filtering and limiting similarity of best fitting models in a Ternary plot, similar to Figure 3 in van der Plas et al. 2015

Usage

```
TernPlot(output)
```

Arguments

output A list with the following elements:
 - DA a vector containing the posterior distribution for the parameter dispersal assembly
 - HF a vector containing the posterior distribution for the parameter habitat filtering
 - LS a vector containing the posterior distribution for the parameter limiting similarity
 This list is generated as output by the function STEPCAM_ABC

Author(s)

Fons van der Plas. Adapted from the ternaryplot function from the 'vcd' package

References

Fons van der Plas, Thijs Janzen, Alejandro Ordonez, Wimke Fokkema, Josephine Reinders, Rampal S. Etienne, and Han Olf 2015. A new modeling approach estimates the relative importance of different community assembly processes. *Ecology* 96:1502-1515.

Examples

```
## Not run:
Artificial.Data <- generate.Artificial.Data(n_species = 40, n_traits = 3,
                                           n_communities = 5, occurrence_distribution = 0.2,
                                           average_richness = 0.5, sd_richness = 0.2,
                                           mechanism_random=FALSE)
O <- STEPCAM_ABC(Artificial.Data$abundances, Artificial.Data$traits,
                 numParticles = 10, n_traits = 3, plot_number = 1, stopRate = 0.8);
TernPlot(O);

## End(Not run)
```

Index

- * **ABC**
 - plotSMC, [4](#)
 - * **SMC**
 - plotSMC, [4](#)
 - * **STEPCAM**
 - STEPCAM-package, [2](#)
 - * **assembly**
 - plotSMC, [4](#)
 - STEPCAM_ABC, [6](#)
 - * **community assembly**
 - STEPCAM-package, [2](#)
 - * **community**
 - plotSMC, [4](#)
 - STEPCAM_ABC, [6](#)
 - * **diversity**
 - STEPCAM_ABC, [6](#)
 - * **functional diversity**
 - STEPCAM-package, [2](#)
 - * **functional**
 - STEPCAM_ABC, [6](#)
 - * **limiting similarity**
 - STEPCAM-package, [2](#)
 - * **processes**
 - STEPCAM_ABC, [6](#)
 - * **simulation**
 - STEPCAM_ABC, [6](#)
 - * **stochasticity**
 - STEPCAM-package, [2](#)
 - * **traits**
 - STEPCAM_ABC, [6](#)
- [generate.Artificial.Data](#), [3](#)
- [plotSMC](#), [4](#)
[plotSTEPCAM](#), [5](#)
- [STEPCAM-package](#), [2](#)
[STEPCAM_ABC](#), [6](#)
- [TernPlot](#), [8](#)