Package ‘DivE’

February 3, 2020

Type Package

Title Diversity Estimator

Version 1.2

Date 2020-02-03

Author Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

Maintainer Daniel Laydon <d.laydon@imperial.ac.uk>

Depends deSolve, FME, rgeos, sp, R (>= 2.15.3)

Description Contains functions for the 'DivE' estimator <doi:10.1371/journal.pcbi.1003646>. The 'DivE' estimator is a heuristic approach to estimate the number of classes or the number of species (species richness) in a population.

License GPL (>= 2)

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-03 17:50:06 UTC

R topics documented:

DivE-package 2
Bact1 4
CombDM 6
Curvature 7
DiveMaster 8
DivSampleNum 11
DivSubsamples 13
FitSingleMod 14
ModelSet 17
ParamRanges 18
ParamSeeds 18
PopDiversity 19
ScoreSingleMod 20

Index **23**

`DivE-package` *Diversity Estimator*

Description

R-package *DivE* contains functions for the DivE estimator (Laydon, D.J. *et al.*, Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014). The *DivE* estimator is a heuristic approach to estimate the number of classes or the number of species (species richness) in a population.

DivE fits many mathematical models to multiple nested subsamples of individual-based rarefaction curves. These curves depict the expected number of species as a function of the number of individuals (e.g. T cells, virions, microbes). Each model is fitted to all nested subsamples, producing multiple model fits. Novel criteria are used to score each model in how consistently its fits reproduce the full observed rarefaction curve from the nested subsamples, i.e. from only incomplete data. The best performing models are extrapolated to a desired population size, and their estimates are aggregated to estimate the number of classes in the population.

The package contains:

1. functions to generate individual-based rarefaction (species-accumulation) data, and evaluate their curvature

2. functions to fit mathematical models to rarefaction data and nested subsamples thereof. These functions make extensive use of the R-package *FME* (http://cran.r-project.org/web/packages/FME/index.html)

3. functions to evaluate novel criteria for each model. These functions make use of the R-package *rgeos* (http://cran.r-project.org/web/packages/rgeos/index.html)

4. functions to score competing models

5. a function to produce final estimates of the number of classes (diversity)

6. example candidate models, fitted parameters, parameter ranges, and an example data set

7. an example script. We have attempted to make the code flexible to users who require varying levels of detail and control. The simplest way to use the package is the DiveMaster function. This function is a wrapper around other functions provided with the *DivE* package and will create subsamples (function `DivSubsamples`), fit models (function `FitSingleMod`), score models (function `ScoreSingleMod`) and produce final diversity/species richness estimates (function `PopDiversity`).

The novel criteria against which each model fit is scored are:

Discrepancy – the mean percentage error between data points and model prediction.

Accuracy – the percentage error between the full sample species richness, and the estimate of full sample species richness from a given subsample.

Similarity – the area between the curve fitted to a subsample and the curve fitted to the full sample, normalized to the area under the curve from the full data, on the interval [0, *Nobs*], where *Nobs* is the size of the full data.

Plausibility – the predicted number of species must either increase monotonically or plateau and the predicted rate of species accumulation must either decrease or plateau (i.e. for $S(x)$ and $x \geq 1$, where x is the number of individuals, $S'(x) \geq 0$, and $S''(x) \leq 0$).

The rationale behind each criterion is as follows:

Discrepancy – the model must describe the data to which it was fitted.

Accuracy – from a subsample, the model should predict the full sample species richness.

Similarity – an ideal model will produce identical fits from all subsamples. The smaller the area between the model fits, the better the model.

Plausibility – this criterion requires that, as the observed number of individuals increases, the observed number of species does not decrease and the rate of species-accumulation does not increase; the former is impossible and the latter is implausible.

Population Size

DivE requires an estimate of population size, i.e. the number of individuals in the population for which the number of species is desired. Population size is a necessary input for species richness estimation when it is not appropriate to assume a saturating relationship between population size and species richness.

In spatially homogeneous populations with equiprobable detection of individuals, population size can be estimated through scaling by area or volume e.g. scaling from cells in 50ml of blood to cells in the total blood volume. When population size estimates are unavailable, it is still usually possible to provide meaningful diversity estimates, e.g. the number of species per gram of tissue.

Requirements

Many deep sequencing data consist of relative abundance of classes or species. We caution that *DivE* requires data detailing the absolute counts of each class or species: relative abundances are insufficient. Rarefaction curves are highly sensitive to the scaling factor applied to relative abundances. Scaling factors that are too high greatly overestimate the degree of repetition of species in the sample, falsely implying that the sample contains a more comprehensive census, and ultimately affecting the resulting estimates of species richness. Absolute counts can usually be obtained when data are being collected (for further details, please see Laydon, D. *et al.*).

DivE requires data where each individual has been sampled randomly, independently and with an equal probability of detection, and where the underlying distribution of individuals is spatially homogeneous. Reliable extrapolation of rarefaction curves is only possible where these conditions are met. DivE is a heuristic estimator designed for use in immunological and microbiological populations, but can be used in any system where the above conditions are satisfied, and for which an estimate of population size is available (for further details, please see Laydon, D. *et al.*).

We have attempted to identify conditions under which DivE is prone to error and should not be applied. When the observed rarefaction curve is linear, the data imply a constant rate of species accumulation, and so provide little information on how quickly the rate of species accumulation will decrease. This is usually indicative of severe under-sampling. We quantified the deviation from linearity of the observed rarefaction curve using the curvature parameter Cp. This parameter can take values between 0 and 1, where 1 reflects perfect saturation and 0 reflects a constant rate of species accumulation. We recommend, based on our simulations, that DivE should not be applied when $Cp < 0.1$. Low curvatures suggest severe under-sampling and researchers should exercise caution when using any diversity estimator with such data. We have included a function `Curvature` to evaluate the approximate curvature of the rarefaction curve.

Model Fitting Process

The pseudo-random model fitting algorithm included with *DivE* (from R package *FME*) requires that parameter ranges and parameter seeding values be inputted. The runtime incurred in model fitting increases with the size of the parameter space. The need for parameter ranges small enough to yield precise parameter estimates in relatively short runtimes must be balanced against the need

for parameter ranges that adequately encompass appropriate parameter values for data of different scales. We have included parameter ranges and seeding values that have performed well in our analyses, which the user can use or amend as required.

The performance of `modFit` (package *FME*) with the pseudorandom parameter search algorithm (package *FME*, `pseudoOptim`) used to estimated model parameter values, is sensitive to the choice of initial seeding values. We have provided the fitted parameters returned from our simulations to be used as initial seeding parameters. For each model, each initial parameter guess (i.e. each row of the model matrix in ModelSeeds) is evaluated by to `modCost`. The parameter guess returning the lowest cost is used as the seeding value in *modFit*.

To obtain better parameter fits, the fitting process can be repeated. Fitted parameters from a single subsample may provide a better seeding guess for a fit to a subsequent subsample than the initial parameter seeds originally inputted, and thus better final model fits will be produced. In our analyses, two attempts of the fitting process (argument `fitloops` in `DiveMaster`) were usually sufficient.

Contact

Daniel J. Laydon, Section of Immunology, Division of Infectious Diseases, Department of Medicine, Imperial College London, Wright-Fleming Institute, Norfolk Place, W2 1PG

d.laydon@imperial.ac.uk

Details

Package:	DivE
Type:	Package
Version:	1.1
Date:	2019-08-17
License:	GPL (>= 2)

The main function is DiveMaster, which combines the four functions DivSampleNum, DivSubsamples, FitSingleMod, and ScoreSingleMod. An example script using both DiveMaster and the four component functions can be found in the *demo* folder in the source.

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

`Bact1` *Count of Medically Important Bacteria Species in a Sample*

Description

This gives a fictitious example of a sample of 7814 bacteria comprising of 144 unique species. Designed as a test dataset for the DivE diversity estimation algorithm.

Usage

```
data(Bact1)
```

Format

A data frame with 144 observations on the following 2 variables.

`Bacteria` a factor with levels `Acetobacter_aurantius` `Acinetobacter_baumannii` `Actinomyces_israelii` `Agrobacterium_radiobacter` `Agrobacterium_tumefaciens` `Anaplasma` `Azorhizobium_caulinodans` `Azotobacter_vinelandii` `Bacillus_anthracis` `Bacillus_brevis` `Bacillus_cereus` `Bacillus_fusiformis` `Bacillus_licheniformis` `Bacillus_megaterium` `Bacillus_mycoides` `Bacillus_stearothermophilus` `Bacillus_subtilis` `Bacteroides_fragilis` `Bacteroides_gingivalis` `Bacteroides_melaninogenicus` `Bartonella_henselae` `Bartonella_quintana` `Bordetella_bronchiseptica` `Bordetella_pertussis` `Borrelia_burgdorferi` `Brucella_abortus` `Brucella_melitensis` `Brucella_suis` `Burkholderia_cepacia` `Burkholderia_mallei` `Burkholderia_pseudomallei` `Calymmatobacterium_granulomatis` `Campylobacter_coli` `Campylobacter_fetus` `Campylobacter_jejuni` `Campylobacter_pylori` `Chlamydia_trachomatis` `Chlamydophila_pneumoniae` `Chlamydophila_psittaci` `Clostridium_botulinum` `Clostridium_difficile` `Clostridium_perfringens` `Clostridium_tetani` `Corynebacterium_diphtheriae` `Corynebacterium_fusiforme` `Coxiella_burnetii` `Ehrlichia_chaffeensis` `Enterobacter_cloacae` `Enterococcus_avium` `Enterococcus_durans` `Enterococcus_faecalis` `Enterococcus_faecium` `Enterococcus_galllinarum` `Enterococcus_maloratus` `Escherichia coli` `Francisella tularensis` `Fusobacterium_nucleatum` `Gardnerella_vaginalis` `Haemophilus_ducreyi` `Haemophilus_influenzae` `Haemophilus_parainfluenzae` `Haemophilus_pertussis` `Haemophilus_vaginalis` `Helicobacter_pylori` `Klebsiella_pneumoniae` `Lactobacillus_Bulgaricus` `Lactobacillus_acidophilus` `Lactobacillus_casei` `Lactococcus_lactis` `Legionella_pneumophila` `Listeria_monocytogenes` `Methanobacterium_extroquens` `Microbacterium_multiforme` `Micrococcus_luteus` `Moraxella_catarrhalis` `Mycobacterium` `Mycobacterium_avium` `Mycobacterium_bovis` `Mycobacterium_diphtheriae` `Mycobacterium_intracellulare` `Mycobacterium_leprae` `Mycobacterium_lepraemurium` `Mycobacterium_phlei` `Mycobacterium_smegmatis` `Mycobacterium_tuberculosis` `Mycoplasma_fermentans` `Mycoplasma_genitalium` `Mycoplasma_hominis` `Mycoplasma_penetrans` `Mycoplasma_pneumoniae` `Neisseria_gonorrhoeae` `Neisseria_meningitidis` `Pasteurella_multocida` `Pasteurella_tularensis` `Peptostreptococcus` `Porphyromonas_gingivalis` `Pseudomonas_aeruginosa` `Rhizobium_radiobacter` `Rickettsia_prowazekii` `Rickettsia_psittaci` `Rickettsia_quintana` `Rickettsia_rickettsii` `Rickettsia_trachomae` `Rochalimaea` `Rochalimaea_henselae` `Rochalimaea_quintana` `Rothia_dentocariosa` `Salmonella_enteritidis` `Salmonella_typhi` `Salmonella_typhimurium` `Serratia_marcescens` `Shigella_dysenteriae` `Staphylococcus_aureus` `Staphylococcus_epidermidis` `Stenotrophomonas_maltophili` `Streptococcus_agalactiae` `Streptococcus_avium` `Streptococcus_bovis` `Streptococcus_cricetus` `Streptococcus_faceium` `Streptococcus_faecalis` `Streptococcus_ferus` `Streptococcus_gallinarum` `Streptococcus_lactis` `Streptococcus_mitior` `Streptococcus_mitis` `Streptococcus_mutans` `Streptococcus_oralis` `Streptococcus_pneumoniae` `Streptococcus_pyogenes` `Streptococcus_rattus` `Streptococcus_salivarius` `Streptococcus_sanguis` `Streptococcus_sobrinus` `Treponema_denticola` `Treponema_pallidum` `Vibrio_cholerae` `Vibrio_comma` `Vibrio_parahaemolyticus` `Vibrio_vulnificus` `Wolbachia` `Yersinia_enterocolitica` `Yersinia_pestis` `Yersinia_pseudotuberculosis`

`Count` a numeric vector

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
data(Bact1)

hist(Bact1[,2], breaks=20, main="Bacterial diversity of a sample",
xlab="Number of bacteria of a given species", ylab="Number of bacterial species")
```

CombDM *CombDM*

Description

Implements the DivE diversity estimator. Combines multiple objects of class *DiveMaster*.

Usage

```
CombDM(dmlist)
```

Arguments

`dmlist`	list of objects of class *DiveMaster*.

Details

CombDM combines multiple objects of class *DiveMaster*. Function used if *DivE* estimation has been split into multiple, separate calls to `DiveMaster`.

Value

An object of class *DiveMaster*, i.e. a list of objects

`model.score`	a matrix of aggregated model scores
`fmm`	a list of *fitsingMod* objects corresponding to the list of fitted models
`ssm`	a matrix of scores of the fit of the models corresponding to the list of fitted models
`estimate`	the estimate of species richness (number of species/classes or diversity) at population size `tot.pop`. This is the geometric average of the models corresponding to the top-five model scores. This is recalculated based on the combined list of models
`lower_estimate`	as per estimate value, but the lowest prediction amongst the models having the top-five scores. This is recalculated based on the combined list of models

`upper_estimate`
: as per estimate value, but the highest prediction amongst the models having the top-five scores. This is recalculated based on the combined list of models

`models`
: list of original input models

`m`
: number of topscoring models used for diversity estimate. This is set as the smallest `m` value of each of the *DiveMaster* objects in the list

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

`DiveMaster`

Examples

```
# See DiveMaster documentation for examples.
```

`Curvature` *Curvature*

Description

Calculates the curvature of the rarefaction curve of the full observed data.

Usage

```
Curvature(dss)
```

Arguments

`dss`
: list of objects of class *DivSubsamples*.

Details

`Curvature` calculates the curvature of the full observed data. If `dss` contains more than one subsample (i.e. if length(`dss`)>1), the curvature of the largest subsample is calculated. If the curvature value is < 0.1, researchers should exercise caution as this is indicative of severe under-sampling, in which case DivE is prone to error.

Value

numeric, between 0 and 1

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

DivSubsamples

Examples

```
# See DivSubsamples documentation for examples.
```

DiveMaster *DiveMaster*

Description

Implements the DivE diversity estimator.

Usage

```
DiveMaster(models, init.params, param.ranges, main.samp,
           tot.pop=(100*(DivSampleNum(main.samp,2)[1])), numit=10^5,
           varleft=1e-8, subsizes=6, dssamps=list(), nrf=1, minrarefac=1,
           NResamples=1000, minplaus=10,
           precision.lv=c(0.0001, 0.005, 0.005), plaus.pen=500,
           crit.wts=c(1.0, 1.0, 1.0, 1.0), fitloops=2, numpred=5)
```

Arguments

`models`	list of models; each model is written as a function: function(x, params) { with(as.list(params), <*function of params*>)}. Examples are given in the ModelSet data file as part of the DivE package.
`init.params`	list of matrices of initial seed model parameters. For each matrix, each row represents a given parameter set; each column represents a parameter value. Column names must match parameter names (params) in the corresponding model in the list `models`. Examples are given in the ParamSeeds data file as part of the DivE package.

`param.ranges` list of matrices of lower and upper model parameters bounds. Used for the modFit function. The first and second row corresponds to the lower and upper bounds respectively; each column represents a parameter value. Column names must match parameter names (params) in the corresponding model in the list `models`. Examples are given in the param.ranges data file as part of the DivE package.

`main.samp` the main sample, either as a 2-column data.frame (species ID, count of species), or a vector of species IDs.

`tot.pop` total population (integer); default set to 100x the `main.samp` size.

`numit` control argument passed to optimisation routine; the maximum number of iterations that modFit will perform. See `modFit` for details.

`varleft` control argument passed to optimisation routine; see `modFit` for details.

`subsizes` either number of nested subsamples (integer, must be 2 or greater), or a vector of nested sample lengths. If the former, then the vector of sample lengths will be created using the *DivSampleNum* function.

`dssamps` list of user specified rarefaction data *DivSubsamples* objects. The length of each component vector of each object in the list must correspond to the vector of nested sample lengths (as defined by the user in `subsizes`).

`nrf` difference between lengths of successive rarefaction datapoints.

`minrarefac` minimum rarefaction x-axis value. This argument is not used if list of *DivSubsamples* object is specified in `dssamps`.

`NResamples` number of resamples used to calculate the rarefaction data. This parameter is not used if list of *DivSubsamples* object is specified in `dssamps`.

`minplaus` lower x-axis bound for plausibility check.

`precision.lv` vector of precision level values for each criterion: 1. discrepancy – mean percentage error between rarefaction data points and model predicion, 2. Sample accuracy – percentage error between observed diversity of full rarefaction data and estimated diversity of full data from subsample, 3. local similarity. The scores for each criteria are defined as 1 + (multiples of bin sizes)

`plaus.pen` penalty score for breaking the plausibility criterion: a model fit should be monotonically increasing and should have a slowing rate of species accumulation.

`crit.wts` vector of weights of each of the four scoring criteria – fit, accuracy, similarity, plausibility. Default is c(1,1,1,1).

`fitloops` number of fitting rounds performed for each model. In each round of fitting, the initial seed parameter values for each model will be the fitted parameters of the previous fitting run. This parameter has a significant impact on the computational time. The 'sweet spot' is 2.

`numpred` number of topscoring models used for diversity prediction. Default is 5.

Details

This is the master function of the DivE estimator. The default operation is a combination of four steps. 1. Generate a list of nested samples lengths from the main sample. 2. For each nested subsample, generate a vector of rarefaction data and their associated mean species diversity. 3. Fit

to the generated data a set of models. 4. Evaluate the fits according to the DivE diversity estimation methodology and compare the scores across models and fitting criteria.

A list of DiveMaster objects, each representing the fits to different sets of models, can combined into a single DiveMaster object using the `CombDM` function. This is useful when running the DivE estimator with the full set of 58 models in a single run is not possible.

One can estimate the diversity for a given population using the `PopDiversity` function where the arguments are the Divemaster object and the population size respectively.

Value

A list of objects:

`model.score`	a matrix of aggregated model scores
`fmm`	a list of *fitsingMod* objects corresponding to the list of fitted models
`ssm`	a matrix of scores of the fit of the models corresponding to the list of fitted models
`estimate`	the estimate of species richness (number of species/classes or diversity) at population size `tot.pop`. This is the geometric average of the models corresponding to the top-five model scores
`lower_estimate`	as per estimate value, but the lowest prediction amongst the models having the top-five scores
`upper_estimate`	as per estimate value, but the highest prediction amongst the models having the top-five scores
`models`	list of original input models
`m`	number of topscoring models used for diversity estimate

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

`FitSingleMod`, `ScoreSingleMod`

Examples

```
require(DivE)
data(Bact1)
data(ModelSet)
data(ParamSeeds)
data(ParamRanges)

testmodels <- list()
testmeta <- list()
paramranges <- list()

# Choose a single model
testmodels <- c(testmodels, ModelSet[1])
#testmeta[[1]] <- (ParamSeeds[[1]]) # Commented out for sake of brevity)
testmeta[[1]] <- matrix(c(0.9451638, 0.007428265, 0.9938149, 1.0147441, 0.009543598, 0.9870419),
                nrow=2, byrow=TRUE, dimnames=list(c(), c("a1", "a2", "a3"))) # Example seeds
paramranges[[1]] <- ParamRanges[[1]]


# Create DivSubsamples object (NB: For quick illustration only -- not default parameters)
dss_1 <- DivSubsamples(Bact1, nrf=2, minrarefac=1, maxrarefac=40, NResamples=5)
dss_2 <- DivSubsamples(Bact1, nrf=2, minrarefac=1, maxrarefac=65, NResamples=5)
dss <- list(dss_2, dss_1)

# Implement the function (NB: For quick illustration only -- not default parameters)
out <- DiveMaster(models=testmodels, init.params=testmeta, param.ranges=paramranges,
                  main.samp=Bact1, subsizes=c(65, 40), NResamples=5, fitloops=1,
                  dssamp=dss, numit=2, varleft=10)

# DiveMaster Outputs
out
out$estimate

out$fmm$logistic

out$fmm$logistic$global

out$ssm

summary(out)

## Combining two DiveMaster objects (assuming a second object 'out2'):
# out3 <- CombDM(list(out, out2))

## To calculate the diversity for a different population size
# PopDiversity(dm=out, popsize=10^5, TopX=1)
```

DivSampleNum *DivSampleNum*

Description

Function to generate an integer sequence representing the lengths of nested samples of sample

Usage

```
DivSampleNum(ms, n)
```

Arguments

`ms`	the main sample, either as a 2-column data.frame (species ID, count of species), or a vector of species IDs.
`n`	desired number of nested samples (integer)

Details

This function produces the default list of nested sample lengths for the DivE algorithm. For the vector representation of the main sample (*ms*) it is equivalent to *sort(round(seq(from=length(ms)/n, to=length(ms), by=length(ms)/n)), decreasing=TRUE)*.

Value

A decreasing sequence of nested sample lengths.

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
require(DivE)
data(Bact1)

DivSampleNum(Bact1, 3)
DivSampleNum(Bact1, 6)
```

DivSubsamples *DivSubsamples*

Description

Function to generate the rarefaction data from a given sample

Usage

```
DivSubsamples(mainsamp, nrf, minrarefac=1,
maxrarefac=length(FormatInput(mainsamp)), NResamples=1000)
```

Arguments

mainsamp	the main sample, either as a 2-column data.frame (species ID, count of species), or a vector of species IDs.
nrf	difference between lengths of successive rarefaction datapoints.
minrarefac	minimum rarefaction data x-axis value. Default is 1.
maxrarefac	maximum rarefaction data x-axis value. Default is length of the sample mainsamp.
NResamples	number of resamples used to calculate the rarefaction data.

Details

This function produces a vector of subsamples diversity values with subsample lengths evenly distributed between a specified minimum and maximum number. The curvature of the rarefaction curve can be obtained with the function Curvature.

Value

a list of class *DivSubsamples* containing resampling results (i.e. the diversity data). This includes the following:

RarefacXAxis	vector of x-axis rarefaction data
RarefacYAxis	vector of y-axis rarefaction data
div_sd	vector of y-axis rarefaction data standard deviations
NResamples	number of sampling iterations used to calculate sample means of each subsample diversity

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
require(DivE)
data(Bact1)

dss_1 <- DivSubsamples(Bact1, nrf=2, minrarefac=1, maxrarefac=100,
                       NResamples=10)
dss_2 <- DivSubsamples(Bact1, nrf=20, minrarefac=1, maxrarefac=100,
                       NResamples=10)
# Default NResamples=1000; low value of NResamples=10 is a set for quick evaluation

dss_1
dss_2

summary(dss_1)
dss_1$div_sd
dss_1$NResamples

Curvature(dss_1)
```

`FitSingleMod` *FitSingleMod*

Description

Function to fit a model to the diversity values of subsamples of a given sample and its nested samples.

Usage

```
FitSingleMod(model.list, init.param, param.range,
             main.samp, tot.pop=(100*(DivSampleNum(main.samp,2)[1])),
             numit=10^5, varleft=1e-8, data.default=TRUE,
             subsizes = 6, dssamps = list(), nrf = 1,
             minrarefac=1, NResamples=1000, minplaus=10,
             fitloops=2)
```

Arguments

`model.list` model; written as a function: function(x, params) with(as.list(params), <*function of params*>). Examples are given in the ModelSet data file as part of the DivE package. Used in the modFit function.

`init.param` matrix of of initial seed model parameters. For each matrix, each row represents a given parameter set; each column represents a parameter value. Column names must match parameter names (params) in the corresponding model in the list `models`. Examples are given in the ParamSeeds data file as part of the DivE package.

`param.range` matrix of lower and upper model parameters bounds. Used for the modFit function. The first and second row corresponds to the lower and upper bounds respectively; each column represents a parameter value. Column names must match parameter names (params) in the corresponding model in the list `models`. Examples are given in the ParamRanges data file as part of the DivE package.

`main.samp` the main sample, either as a 2-column data.frame (species ID, count of species), or a vector of species IDs.

`tot.pop` total population (integer); default set to 100x the `main.samp` size.

`numit` control argument passed to optimisation routine; the maximum number of iterations that modFit will perform. See `modFit` for details.

`varleft` control argument passed to optimisation routine; see `modFit` for details.

`data.default` if `True`, then the list of vectors of nested rarefaction data (divsubsample objects) generated by the *DivSampleNum* and *divsubsample* functions; if `False`, then the function uses the user-specified list of nested rarefaction data, *dssamps*

`subsizes` either number of subsamples of main.samp (integer), or a vector of subsample lengths. If the former, then the vector of sample lengths will be created using the *DivSampleNum* function.

`dssamps` list of user specified rarefaction data *DivSubsamples* objects. The length of each component vector of each object in the list must correspond to the vector of subsample lengths (as defined by the user in `subsizes`).

`nrf` difference between lengths of successive rarefaction datapoints.

`minrarefac` minimum rarefaction x-axis value. This argument is not used if list of *DivSubsamples* object is specified in *dssamps*.

`NResamples` number of resamples used to calculate the rarefaction data. This parameter is not used if list of *DivSubsamples* object is specified in `dssamps`. NB: different from `numit` parameter, which is specific to the fitting process.

`minplaus` lower x-axis bound for plausibility check.

`fitloops` number of fitting rounds performed for each model. In each round of fitting, the initial seed parameter values for each model will be the fitted parameters of the previous fitting run. This parameter has a significant impact on the computational time. The 'sweet spot' is 2.

Details

This function fits a single specified model to the diversity values of the subsamples of a set of nested samples. The output is a list of raw fitting results (pre-scoring). The user should use this function if he or she is interested in fitting a specific parametric rarefraction curve to a sample (rather than selecting the most appropriate model) and examining its performance.

Value

A list of class *FitSingleMod* containing the results of the fit of the model to the diversity samples. This includes the following:

`param` matrix of fitted parameters for each nested sample

`ssr`	sum-of-squared residuals for the fits for each nested sample
`ms`	mean sum-of-squared residuals for the fits for each nested sample
`discrep`	goodness-of-fit values for the fits for each nested sample; this expressed as the average across the subsamples in each nested sample of all the percentage residuals
`local`	prediction of main sample sizes according to fitted curves for each of the nested samples
`global`	prediction of population diversity at `popsize` according to fitted curves for each of the nested subsamples
`AccuracyToObserved`	vector of percentage errors between the observed diversity of full sample data and the estimated diversity of full sample data from subsamples
`subsamplesizes`	vector of nested subsample sizes
`datapoints`	the list of *divsubsample* objects used in the fitting. The length of the list is equal to number of samples
`modelname`	name of the model used
`numparam`	number of parameters in the model
`sampvar`	the mean squared distances between subsample curves, local and global
`mono.local`	matrix of logical values: is the curve monotonically increasing, up to the main sample size?
`mono.global`	matrix of logical values: is the curve monotonically increasing, up to the population size?
`slowing.local`	matrix of logical values: is the rate of increase in the curve slowing (decreasing second derivative), up to the main sample size?
`slowing.global`	matrix of logical values: is the rate of increase in the curve slowing (decreasing second derivative), from `minplaus` to the population size (`popsize`?
`plausibility`	matrix of logical values: is the curve plausible (i.e. monotonically increasing and with decreasing second derivative)?
`dist.local`	matrix of distances between curves fitted to the nested samples. Distances are calculated as areas between curves bounded by 0 and the main sample size
`dist.global`	similar to *dist.local*, but with curve upper bound the population size
`local.ref.dist`	distances of nested curves to the curve fitted to the whole sample, with the curves bounded by 0 and the main sample size
`global.ref.dist`	similar to *local.ref.dist* but with curve upper bound the population size
`popsize`	user defined population size
`the model`	the function corresponding to the user-selected `modelname`

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

ScoreSingleMod

Examples

```
# See documentation of \code{ScoreSingleMod} for examples
```

ModelSet — *List of 58 candidate models to fit to data*

Description

ModelSet is an example list of candidate models used in the reference below to calculate the *DivE* estimate

Usage

```
data(ModelSet)
```

Format

A list of 58 named functions (with named parameters). Each model in the list must be provided as a function, and must be of the following form: function(x, params) with(as.list(params), <*function of params*>). The parameter names are a1, a2, a3, etc. These must match the names of the parameter values given in *ParamSeeds* and *ParamRanges*.

Details

Each model is written as a function: function(x, params) with(as.list(params), <*function of params*>). Examples are given in the *ModelSet* data file as part of the *DivE* package. The user can amend *ModelSet* and input additional models as required. The analytical form of all the models provided in *ModelSet* can be found in the reference below, in Text S1: List of *DivE* candidate models. All models were obtained from *zunzun.org*, an online curve fitting repository

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
data(ModelSet)
```

ParamRanges — *List of 58 sets of upper and lower bounds for models evaluated by DivE*

Description

A list of 58 matrices. Each matrix corresponds to a model in `ModelSet`, for which it contains suggested upper and lower bounds for each parameter.

Usage

```
data(ParamRanges)
```

Format

A list of 58 matrices. Each matrix has 2 rows (lower bounds, upper bounds) and columns corresponding to the parameters of the matching model in `ModelSet`.

Details

There is a trade-off between specifying parameter ranges that are large enough to encompass likely fitted values for a variety of data sets, and specifying parameter ranges that are suitably small so that parameter estimation is sufficiently precise and runtime is managable. We have aimed to balance these competing concerns. The parameter ranges provided performed well in our simulations. The user can amend if required.

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
data(ParamRanges)
```

ParamSeeds — *List of 58 matrices of model seeding parameters.*

Description

The performance of *modFit* (package *FME*) with the pseudorandom parameter search algorithm (package *FME*, `pseudoOptim`) used to estimated model parameter values, is sensitive to the choice of initial seeding values. We have provided the fitted parameters returned from our simulations to be used as initial seeding parameters.

Usage

```
data(ParamSeeds)
```

Format

A list of 58 matrices. Each matrix has columns corresponding to the parameters of the matching model in *ModelSet*. Each row is set of potential seeding parameters

Details

For each model, each initial parameter guess (i.e. each row of the model matrix in *ParamRanges*) is evaluated by to *modCost*. The parameter guess returning the lowest cost is used as the seeding value in *modFit*. If the user wishes to input alternative initial seeding parameter values, then for each model and parameter, all values must be finite (not `NA` or `NaN`), and within the upper and lower bounds set in *ParamRanges*. Column names must match parameter names (params) in the corresponding model in ModelSet (i.e. `models` argument in DivEMaster).

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

Examples

```
data(ParamSeeds)
```

`PopDiversity` *PopDiversity*

Description

Calculates the species richness at a specified population size, taking an object of class *DiveMaster* as an input.

Usage

```
PopDiversity(dm, popsize, TopX=NULL)
```

Arguments

`dm`	list of objects of class *DiveMaster*.
`popsize`	positive real number. Population size.
`TopX`	a positive integer, less than the number of models contained in `dm`, representing the number of best-performing models used for the aggregated estimate of the population diversity. If `NULL` (default), then `dm$m` models are aggregated. If `TopX` is larger than the the number of models fitted, then min(5, length(dm$fmm)) models are aggregated.

Details

CombDM combines multiple objects of class *DiveMaster*. Function used if *DivE* estimation has been split into multiple, separate calls to `DiveMaster`.

Value

A list of objects:

`estimate` point estimate of diversity (species richness)

`upper_estimate` estimate upper bound

`lower_estimate` estimate lower bound

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

`DiveMaster`

Examples

```
# See DiveMaster documentation for examples.
```

`ScoreSingleMod` *ScoreSingleMod*

Description

Determines the set of scores corresponding to a single model fit to a diversity values of subsamples of a given sample and its nested samples.

Usage

```
ScoreSingleMod(fsm, precision.lv=c(0.0001, 0.005, 0.005), plaus.pen=500)
```

Arguments

`fsm`	*FitSingleMod* object
`precision.lv`	vector of precision level values for each criterion: 1. discrepancy – mean percentage error between rarefaction data points and model predicion, 2. Sample accuracy – percentage error between observed diversity of full rarefaction data and estimated diversity of full data from subsample, 3. local similarity. The scores for each criteria are defined as 1 + (multiples of bin sizes)
`plaus.pen`	penalty score for breaking the plausibility criterion: a model fit should be monotonically increasing and should have a slowing rate of species accumulation.

Details

The score for a given model is only meaningful when compared with scores of other models. Lower score = better for predicting the population diversity. To assess the performance of a single model, it is more informative to use `FitSingleMod` function.

Value

A list of class *ScoreSingleMod* containing the scores of the fit of the model to the diversity samples. This includes the following:

`discrepancy`	score for discrepancy, aggregated across all nested subsamples
`accuracy`	score for accuracy of full sample prediction, aggregated across all nested subsamples
`similarity`	score for similarity of curves for different samples
`plausibility`	score for plausibility criterion
`binsize`	vector of user-specified precision values used to translate values associated with each criterion into scores
`plausibility.penalty`	penalty score for implausible diversity curve
`modname`	model name

Author(s)

Daniel J. Laydon, Aaron Sim, Charles R.M. Bangham, Becca Asquith

References

Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, S., Douek, D. C., Price, D., Bangham, C. R. M., Asquith, B., Quantification of HTLV-1 clonality and TCR diversity, PLOS Comput. Biol. 2014

See Also

`FitSingleMod`

Examples

```
require(DivE)
data(Bact1)
data(ModelSet)
data(ParamSeeds)
data(ParamRanges)

testmodels <- list()
testmeta <- list()
paramranges <- list()

# Choose a single model

testmodels <- c(testmodels, ModelSet[1])
# testmeta <- (ParamSeeds[[1]]) # Commented out for sake of brevity)
testmeta <- matrix(c(0.9451638, 0.007428265, 0.9938149, 1.0147441, 0.009543598, 0.9870419),
                nrow=2, byrow=TRUE, dimnames=list(c(), c("a1", "a2", "a3"))) # Example seeds
paramranges <- ParamRanges[[1]]

# Create DivSubsamples object (NB: For quick illustration only -- not default parameters)
dss_1 <- DivSubsamples(Bact1, nrf=2, minrarefac=1, maxrarefac=40, NResamples=5)
dss_2 <- DivSubsamples(Bact1, nrf=2, minrarefac=1, maxrarefac=65, NResamples=5)
dss <- list(dss_2, dss_1)

# Fit the model (NB: For quick illustration only -- not default parameters)
fsm <- FitSingleMod(model.list=testmodels, init.param=testmeta, param.range=paramranges,
                    main.samp=Bact1, dssamps=dss, fitloops=1, data.default=FALSE,
                    subsizes=c(65, 40),
                    numit=2) # numit chosen to be extremely small to speed up example


# Score the model
ssm <- ScoreSingleMod(fsm)

ssm
summary(ssm)
```

Index

∗Topic **datasets**
Bact1, 4
ModelSet, 17
ParamRanges, 18
ParamSeeds, 18
∗Topic **diversity**
CombDM, 6
Curvature, 7
DiveMaster, 8
DivSampleNum, 11
DivSubsamples, 13
FitSingleMod, 14
PopDiversity, 19
ScoreSingleMod, 20
∗Topic **package**
DivE-package, 2

Bact1, 4
Bact2 *(*Bact1*)*, 4

CombDM, 6
Curvature, 7

DivE *(*DivE-package*)*, 2
DivE-package, 2
DiveMaster, *7*, 8, *20*
DivSampleNum, 11
DivSubsamples, *8*, 13

FitSingleMod, *10*, 14, *21*

ModelSet, 17
modFit, *9*, *15*

ParamRanges, 18
ParamSeeds, 18
plot.FitSingleMod *(*FitSingleMod*)*, 14
PopDiversity, 19
print.DiveMaster *(*DiveMaster*)*, 8
print.DivSubsamples *(*DivSubsamples*)*, 13
print.FitSingleMod *(*FitSingleMod*)*, 14
print.ScoreSingleMod *(*ScoreSingleMod*)*, 20
print.summary.DiveMaster *(*DiveMaster*)*, 8
print.summary.DivSubsamples *(*DivSubsamples*)*, 13
print.summary.FitSingleMod *(*FitSingleMod*)*, 14
print.summary.ScoreSingleMod *(*ScoreSingleMod*)*, 20

ScoreSingleMod, *10*, *17*, 20
summary.DiveMaster *(*DiveMaster*)*, 8
summary.DivSubsamples *(*DivSubsamples*)*, 13
summary.FitSingleMod *(*FitSingleMod*)*, 14
summary.ScoreSingleMod *(*ScoreSingleMod*)*, 20