

# Package: mobr (via r-universe)

October 31, 2024

**Title** Measurement of Biodiversity

**Version** 3.0.2

**Date** 2024-08-10

**Description** Functions for calculating metrics for the measurement of biodiversity and its changes across scales, treatments, and gradients. The methods implemented in this package are described in: Chase, J.M., et al. (2018) [doi:10.1111/ele.13151](https://doi.org/10.1111/ele.13151), McGlinn, D.J., et al. (2019) [doi:10.1111/2041-210X.13102](https://doi.org/10.1111/2041-210X.13102), McGlinn, D.J., et al. (2020) [doi:10.1101/851717](https://doi.org/10.1101/851717), and McGlinn, D.J., et al. (2023) [doi:10.1101/2023.09.19.558467](https://doi.org/10.1101/2023.09.19.558467).

**Depends** R (>= 3.5.0)

**Imports** plotrix, scales, dplyr, purrr, tidyr, pbapply, ggplot2, egg, tibble, vctrs, rlang, geosphere, scam, sf

**Suggests** knitr, rmarkdown, testthat, methods

**Language** en-US

**License** MIT + file LICENSE

**LazyData** true

**RoxygenNote** 7.3.1

**URL** <https://github.com/MoBiodiv/mobr>

**BugReports** <https://github.com/MoBiodiv/mobr/issues>

**Encoding** UTF-8

**VignetteBuilder** knitr

**Repository** <https://mobiodiv.r-universe.dev>

**RemoteUrl** <https://github.com/mobiodiv/mobr>

**RemoteRef** HEAD

**RemoteSha** d1c219314d72c7a84621e2425fd38cdba7fba099

## Contents

avg_nn_dist	2
calc_beta_div	3
calc_chao1	6
calc_comm_div	7
calc_comm_div_ci	10
calc_C_target	13
calc_div	13
calc_PIE	15
calc_SPIE	16
calc_S_C	17
Chat	18
compare_samp_rarefaction	19
fire_comm	20
get_delta_stats	20
get_mob_stats	23
get_null_comm	26
get_samples	27
invChat	29
inv_comm	30
kNCN_average	30
make_mob_in	32
mobr	33
plot.mob_out	34
plot.mob_stats	35
plot_abu	36
plot_comm_div	37
plot_N	39
plot_rarefaction	39
rarefaction	41
subset.mob_in	45
tank_comm	46
<b>Index</b>	<b>47</b>

---

avg\_nn\_dist

*Compute average nearest neighbor distance*

---

### Description

This function computes the average distance of the next nearest sample for a given set of coordinates. This method of sampling is used by the function `rarefaction` when building the spatial, sample-based rarefaction curves (sSBR).

### Usage

```
avg_nn_dist(coords)
```

**Arguments**

coords            a matrix with n-dimensional coordinates

**Value**

a vector of average distances for each sequential number of accumulated nearest samples.

**Examples**

```
# transect spatial arrangement
transect = 1:100
avg_nn_dist(transect)
grid = expand.grid(1:10, 1:10)
avg_nn_dist(grid)
oldpar <- par(no.readonly = TRUE)
par(mfrow=c(1,2))
plot(avg_nn_dist(transect), type='o', main='transect',
      xlab='# of samples', ylab='average distance')
# 2-D grid spatial arrangement
plot(avg_nn_dist(grid), type='o', main='grid',
      xlab='# of samples', ylab='average distance')
par(oldpar)
```

---

 calc\_beta\_div

---

*Calculate beta diversity from sites by species table.*


---

**Description**

This function computes multiplicative beta diversity by computing the ratio of gamma diversity to average alpha diversity. It is a wrapper for the function `calc_comm_div` when that function's scales is set to 'betas'.

**Usage**

```
calc_beta_div(
  abund_mat,
  index,
  effort = NA,
  C_target_gamma = NA,
  extrapolate = TRUE,
  ...
)
```

**Arguments**

abund\_mat        Abundance based site-by-species table. Species as columns

index            The calculated biodiversity indices. The options are

- N ... Number of individuals (total abundance)

- S ... Number of species
- S\_n ... Rarefied or extrapolated number of species for n individuals
- S\_C ... Estimate species richness of a given level of coverage by C\_target\_gamma
- S\_asymp ... Estimated asymptotic species richness
- f\_0 ... Estimated number of undetected species
- pct\_rare ... The percent of rare species as defined by rare\_thres
- PIE ... Hurlbert's PIE (Probability of Interspecific Encounter)
- S\_PIE ... Effective number of species based on PIE

See *Details* for additional information on the biodiversity statistics.

effort	The standardized number of individuals used for the calculation of rarefied species richness. This must be a single integer.
C_target_gamma	When computing coverage based richness (S_C) then this argument can be used to specify the coverage to be used for the gamma scale richness estimate. This defaults to NA in which case the target cover is computed by <a href="#">calc_C_target</a> (i.e., the largest allowable sample size).
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method (Chao 1984, 1987). Defaults to TRUE.
...	other arguments to pass to calc_comm_div

## Details

### BIODIVERSITY INDICES

**N: total community abundance** is the total number of individuals observed across all species in the sample

**S: species richness** is the observed number of species that occurs at least once in a sample

**S\_n: Rarefied species richness** is the expected number of species, given a defined number of sampled individuals (n) (Gotelli & Colwell 2001). Rarefied richness at the alpha-scale is calculated for the values provided in `effort_samples` as long as these values are not smaller than the user-defined minimum value `effort_min`. In this case the minimum value is used and samples with less individuals are discarded. When no values for `effort_samples` are provided the observed minimum number of individuals of the samples is used, which is the standard in rarefaction analysis (Gotelli & Colwell 2001). Because the number of individuals is expected to scale linearly with sample area or effort, at the gamma-scale the number of individuals for rarefaction is calculated as the minimum number of samples within groups multiplied by `effort_samples`. For example, when there are 10 samples within each group, `effort_groups` equals  $10 * \text{effort\_samples}$ . If n is larger than the number of individuals in sample and `extrapolate = TRUE` then the Chao1 (Chao 1984, 1987) method is used to extrapolate the rarefaction curve.

**pct\_rare: Percent of rare species** Is the ratio of the number of rare species to the number of observed species  $\times 100$  (McGill 2011). Species are considered rare in a particular sample if they have fewer individuals than  $\text{rare\_thres} * N$  where `rare_thres` can be set by the user and N is the total number of individuals in the sample. The default value of `rare_thres` of 0.05 is arbitrary and was chosen because McGill (2011) found this metric of rarity performed well and was generally less correlated with other common metrics of biodiversity. Essentially this metric attempt to estimate

what proportion of the species in the same occur in the tail of the species abundance distribution and is therefore sensitive to presence of rare species.

**S\_asymp: Asymptotic species richness** is the expected number of species given complete sampling and here it is calculated using the Chao1 estimator (Chao 1984, Chao 1987) see [calc\\_chao1](#). Note: this metric is typically highly correlated with S (McGill 2011).

**f\_0: Undetected species richness** is the number of undetected species or the number of species observed 0 times which is an indicator of the degree of rarity in the community. If there is a greater rarity then f\_0 is expected to increase. This metric is calculated as S\_asymp - S. This metric is less correlated with S than the raw S\_asymp metric.

**PIE: Probability of intraspecific encounter** represents the probability that two randomly drawn individuals belong to the same species. Here we use the definition of Hurlbert (1971), which considers sampling without replacement. PIE is closely related to the well-known Simpson diversity index, but the latter assumes sampling with replacement.

**S\_PIE: Effective number of species for PIE** represents the effective number of species derived from the PIE. It is calculated using the asymptotic estimator for Hill numbers of diversity order 2 (Chao et al. 2014). S\_PIE represents the species richness of a hypothetical community with equally-abundant species and infinitely many individuals corresponding to the same value of PIE as the real community. An intuitive interpretation of S\_PIE is that it corresponds to the number of dominant (highly abundant) species in the species pool.

For species richness S, rarefied richness S\_n, undetected richness f\_0, and the Effective Number of Species S\_PIE we also calculate beta-diversity using multiplicative partitioning (Whittaker 1972, Jost 2007). That means for these indices we estimate beta-diversity as the ratio of gamma-diversity (total diversity across all plots) divided by alpha-diversity (i.e., average plot diversity).

## Value

A data.frame with four columns:

- scale ... Group label for sites
- index ... Name of the biodiversity index
- sample\_size ... The number of samples used to compute the statistic, helpful for interpreting beta and gamma metrics.
- effort ... Sampling effort for rarefied richness (NA for the other indices)
- gamma\_coverage ... The coverage value for that particular effort value on the gamma scale rarefaction curve. Will be NA unless coverage based richness (S\_C) and/or beta diversity is computed.
- value ... Value of the biodiversity index

## Author(s)

Felix May and Dan McGlenn

## References

Chao, A. 1984. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics* 11:265–270.

- Chao, A. 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Gotelli, N. J., and R. K. Colwell. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters* 4:379–391.
- Hurlbert, S. H. 1971. The nonconcept of species diversity: a critique and alternative parameters. *Ecology* 52:577–586.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88:2427–2439.
- McGill, B. J. 2011. Species abundance distributions. Pages 105-122 *Biological Diversity: Frontiers in Measurement and Assessment*, eds. A.E. Magurran and B.J. McGill.
- Whittaker, R. H. 1972. Evolution and measurement of species diversity. *Taxon* 21:213–251.

### See Also

[calc\\_comm\\_div](#)

### Examples

```
data(inv_comm)
beta_metrics = calc_beta_div(inv_comm, 'S_n', effort = c(5, 10))
beta_metrics
```

---

calc_chao1	<i>Estimation of species richness</i>
------------	---------------------------------------

---

### Description

calc\_chao1 estimates the number of species at the asymptote ( $S_{\text{asympt}}$ ) of the species accumulation curve based on the methods proposed in Chao (1984, 1987, 2005).

### Usage

```
calc_chao1(x)
```

### Arguments

x                      a vector of species abundances or a site-by-species matrix

### Details

This function is a trimmed version of `iNext::ChaoRichness`. T. C. Hsieh, K. H. Ma and Anne Chao are the original authors of the iNEXT package.

### Value

a vector of species richness estimates

## References

- Chao, A. (1984) Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of Statistics*, 11, 265-270.
- Chao, A. (1987) Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Chao, A. (2005) Species estimation and applications. Pages 7907-7916 in N. Balakrishnan, C. B. Read, and B. Vidakovic, editors. *Encyclopedia of statistical sciences*. Second edition, volume 12. Wiley, New York, New York, USA.

## Examples

```
data(inv_comm)
calc_chao1(inv_comm)
```

---

calc\_comm\_div

*Calculate biodiversity statistics from sites by species table.*

---

## Description

Calculate biodiversity statistics from sites by species table.

## Usage

```
calc_comm_div(
  abund_mat,
  index,
  effort = NA,
  extrapolate = TRUE,
  return_NA = FALSE,
  rare_thres = 0.05,
  scales = c("alpha", "gamma", "beta"),
  avg_alpha = FALSE,
  PIE_replace = FALSE,
  C_target_gamma = NA,
  ...
)
```

## Arguments

- |           |   |
|-----------|---|
| abund_mat | Abundance based site-by-species table. Species as columns   |
| index     | The calculated biodiversity indices. The options are <ul style="list-style-type: none"> <li>• N ... Number of individuals (total abundance)</li> <li>• S ... Number of species</li> <li>• S_n ... Rarefied or extrapolated number of species for n individuals</li> <li>• S_C ... Estimate species richness of a given level of coverage by C_target_gamma</li> </ul> |

- S\_asymp ... Estimated asymptotic species richness
- f\_0 ... Estimated number of undetected species
- pct\_rare ... The percent of rare species as defined by rare\_thres
- PIE ... Hurlbert's PIE (Probability of Interspecific Encounter)
- S\_PIE ... Effective number of species based on PIE

See *Details* for additional information on the biodiversity statistics.

effort	The standardized number of individuals used for the calculation of rarefied species richness. This must be a single integer.
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method (Chao 1984, 1987). Defaults to TRUE.
return_NA	Boolean in which the rarefaction function returns the observed S when effort is larger than the number of individuals. If set to TRUE then NA is returned. Note that this argument is only relevant when extrapolate = FALSE.
rare_thres	The threshold that determines how the metric pct_rare is computed. It can range from (0, 1] and defaults to 0.05 which specifies that any species with less than or equal to 5 abundance in a sample is considered rare. It can also be specified as "N/S" which results in using average abundance as the threshold which McGill (2011) found to have the best small sample behavior.
scales	The scales to compute the diversity indices for: <ul style="list-style-type: none"> <li>• alpha ... for each row of the site x species community matrix</li> <li>• gamma ... for the entire site x species community matrix</li> <li>• beta ... the ratio of diversity at the gamma and alpha scales.</li> </ul> Defaults to all three scales: c('alpha', 'gamma', 'beta')
avg_alpha	Boolean if TRUE then the alpha values are averaged. Defaults to FALSE.
PIE_replace	Used for PIE and SPIE. If TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).
C_target_gamma	When computing coverage based richness (S_C) then this argument can be used to specify the coverage to be used for the gamma scale richness estimate. This defaults to NA in which case the target cover is computed by <a href="#">calc_C_target</a> (i.e., the largest allowable sample size).
...	additional arguments that can be passed to <a href="#">calc_div</a>

## Details

### BIODIVERSITY INDICES

**N: total community abundance** is the total number of individuals observed across all species in the sample

**S: species richness** is the observed number of species that occurs at least once in a sample

**S<sub>n</sub>: Rarefied species richness** is the expected number of species, given a defined number of sampled individuals (n) (Gotelli & Colwell 2001). Rarefied richness at the alpha-scale is calculated for the values provided in `effort_samples` as long as these values are not smaller than the user-defined minimum value `effort_min`. In this case the minimum value is used and samples with



less individuals are discarded. When no values for `effort_samples` are provided the observed minimum number of individuals of the samples is used, which is the standard in rarefaction analysis (Gotelli & Colwell 2001). Because the number of individuals is expected to scale linearly with sample area or effort, at the gamma-scale the number of individuals for rarefaction is calculated as the minimum number of samples within groups multiplied by `effort_samples`. For example, when there are 10 samples within each group, `effort_groups` equals  $10 * \text{effort\_samples}$ . If `n` is larger than the number of individuals in sample and `extrapolate = TRUE` then the Chao1 (Chao 1984, 1987) method is used to extrapolate the rarefaction curve.

**pct\_rare: Percent of rare species** Is the ratio of the number of rare species to the number of observed species  $\times 100$  (McGill 2011). Species are considered rare in a particular sample if they have fewer individuals than `rare_thres * N` where `rare_thres` can be set by the user and `N` is the total number of individuals in the sample. The default value of `rare_thres` of 0.05 is arbitrary and was chosen because McGill (2011) found this metric of rarity performed well and was generally less correlated with other common metrics of biodiversity. Essentially this metric attempt to estimate what proportion of the species in the same occur in the tail of the species abundance distribution and is therefore sensitive to presence of rare species.

**S\_asymp: Asymptotic species richness** is the expected number of species given complete sampling and here it is calculated using the Chao1 estimator (Chao 1984, Chao 1987) see [calc\\_chao1](#). Note: this metric is typically highly correlated with `S` (McGill 2011).

**f\_0: Undetected species richness** is the number of undetected species or the number of species observed 0 times which is an indicator of the degree of rarity in the community. If there is a greater rarity then `f_0` is expected to increase. This metric is calculated as `S_asymp - S`. This metric is less correlated with `S` than the raw `S_asymp` metric.

**PIE: Probability of intraspecific encounter** represents the probability that two randomly drawn individuals belong to the same species. Here we use the definition of Hurlbert (1971), which considers sampling without replacement. PIE is closely related to the well-known Simpson diversity index, but the latter assumes sampling with replacement.

**S\_PIE: Effective number of species for PIE** represents the effective number of species derived from the PIE. It is calculated using the asymptotic estimator for Hill numbers of diversity order 2 (Chao et al. 2014). `S_PIE` represents the species richness of a hypothetical community with equally-abundant species and infinitely many individuals corresponding to the same value of PIE as the real community. An intuitive interpretation of `S_PIE` is that it corresponds to the number of dominant (highly abundant) species in the species pool.

For species richness `S`, rarefied richness `S_n`, undetected richness `f_0`, and the Effective Number of Species `S_PIE` we also calculate beta-diversity using multiplicative partitioning (Whittaker 1972, Jost 2007). That means for these indices we estimate beta-diversity as the ratio of gamma-diversity (total diversity across all plots) divided by alpha-diversity (i.e., average plot diversity).

## Value

A data.frame with four columns:

- `scale ...` Group label for sites
- `index ...` Name of the biodiversity index
- `sample_size ...` The number of samples used to compute the statistic, helpful for interpreting beta and gamma metrics.

- effort ... Sampling effort for rarefied richness (NA for the other indices)
- gamma\_coverage ... The coverage value for that particular effort value on the gamma scale rarefaction curve. Will be NA unless coverage based richness (S\_C) and/or beta diversity is computed.
- value ... Value of the biodiversity index

### Author(s)

Felix May and Dan McGlenn

### References

- Chao, A. 1984. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics* 11:265–270.
- Chao, A. 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Gotelli, N. J., and R. K. Colwell. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters* 4:379–391.
- Hurlbert, S. H. 1971. The nonconcept of species diversity: a critique and alternative parameters. *Ecology* 52:577–586.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88:2427–2439.
- McGill, B. J. 2011. Species abundance distributions. Pages 105-122 *Biological Diversity: Frontiers in Measurement and Assessment*, eds. A.E. Magurran and B.J. McGill.
- Whittaker, R. H. 1972. Evolution and measurement of species diversity. *Taxon* 21:213–251.

### Examples

```
data(tank_comm)
div_metrics <- calc_comm_div(tank_comm, 'S')
div_metrics
div_metrics <- calc_comm_div(tank_comm, 'S', avg_alpha = TRUE)
div_metrics
div_metrics <- calc_comm_div(tank_comm, 'S_n', effort = 10, avg_alpha = TRUE)
div_metrics
div_metrics <- calc_comm_div(tank_comm, 'S_C', C_target_gamma = 0.75, avg_alpha = TRUE)
div_metrics
```

---

calc\_comm\_div\_ci

*Compute non-parametric confidence intervals across diversity indices.*

---

### Description

This function take a list of community matrices and returns the central tendency and the confidence interval for each diversity index of interest across that list of communities.

**Usage**

```
calc_comm_div_ci(
  samples,
  cent_stat = "median",
  ci = c(0.025, 0.975),
  index,
  effort = NA,
  extrapolate = TRUE,
  return_NA = FALSE,
  rare_thres = 0.05,
  scales = c("alpha", "gamma", "beta"),
  PIE_replace = FALSE,
  C_target_gamma = NA,
  ...
)
```

**Arguments**

samples	a list of community matrices (i.e., the output of <code>get_samples</code> )
cent_stat	a string that is either 'mean' or 'median' which specifies the measure of central tendency. Defaults to 'median'.
ci	a numeric vector of two numbers specifying the lower and upper quantiles of the distribution to return. The default is to report the 0.025 and 0.975 quantiles in other words a 95 percent interval.
index	The calculated biodiversity indices. The options are <ul style="list-style-type: none"> <li>• N ... Number of individuals (total abundance)</li> <li>• S ... Number of species</li> <li>• S_n ... Rarefied or extrapolated number of species for n individuals</li> <li>• S_C ... Estimate species richness of a given level of coverage by <code>C_target_gamma</code></li> <li>• S_asymp ... Estimated asymptotic species richness</li> <li>• f_0 ... Estimated number of undetected species</li> <li>• pct_rare ... The percent of rare species as defined by <code>rare_thres</code></li> <li>• PIE ... Hurlbert's PIE (Probability of Interspecific Encounter)</li> <li>• S_PIE ... Effective number of species based on PIE</li> </ul> <p>See <i>Details</i> for additional information on the biodiversity statistics.</p>
effort	The standardized number of individuals used for the calculation of rarefied species richness. This must be a single integer.
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method (Chao 1984, 1987). Defaults to TRUE.
return_NA	Boolean in which the rarefaction function returns the observed S when effort is larger than the number of individuals. If set to TRUE then NA is returned. Note that this argument is only relevant when <code>extrapolate = FALSE</code> .

rare_thres	The threshold that determines how the metric <code>pct_rare</code> is computed. It can range from (0, 1] and defaults to 0.05 which specifies that any species with less than or equal to 5 abundance in a sample is considered rare. It can also be specified as "N/S" which results in using average abundance as the threshold which McGill (2011) found to have the best small sample behavior.
scales	The scales to compute the diversity indices for: <ul style="list-style-type: none"> <li>• alpha ... for each row of the site x species community matrix</li> <li>• gamma ... for the entire site x species community matrix</li> <li>• beta ... the ratio of diversity at the gamma and alpha scales.</li> </ul> Defaults to all three scales: <code>c('alpha', 'gamma', 'beta')</code>
PIE_replace	Used for PIE and SPIE. If TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).
C_target_gamma	When computing coverage based richness ( <code>S_C</code> ) then this argument can be used to specify the coverage to be used for the gamma scale richness estimate. This defaults to NA in which case the target cover is computed by <code>calc_C_target</code> (i.e., the largest allowable sample size).
...	additional arguments that can be passed to <code>calc_div</code>

### Details

The measure of central tendency can be the median (default) or mean, and the range of the confidence interval can be specified but defaults to a 95 confidence interval.

### Value

a data.frame that has the lower, middle, and upper quantiles of the sampled distribution of each diversity index.

### See Also

`get_samples` for generating samples, and `calc_comm_div` for the calculation of diversity indices.

### Examples

```
data(tank_comm)
samples <- get_samples(tank_comm, algo = 'loo')
calc_comm_div_ci(samples, index = 'S_PIE')
samples <- get_samples(tank_comm, algo = 'boot', n_boot = 20)
calc_comm_div_ci(samples, index = 'S_PIE')
# compute ci for average diversity (rather than median)
calc_comm_div_ci(samples, index = 'S_PIE', cent_stat = 'avg')
```

---

calc_C_target	<i>Calculate the recommended target coverage value for the computation of beta_C</i>
---------------	--

---

### Description

Returns the estimated gamma-scale coverage that corresponds to the largest allowable sample size (i.e. the smallest observed sample size at the alpha scale multiplied by an extrapolation factor). The default (factor = 2) allows for extrapolation up to 2 times the observed sample size of the smallest alpha sample. For factor= 1, only interpolation is applied. Factors larger than 2 are not recommended.

### Usage

```
calc_C_target(x, factor = 2)
```

### Arguments

x	a site by species abundance matrix
factor	numeric. A multiplier for how much larger than total community abundance to extrapolate to. Defaults to 2.

### Value

numeric value

### Examples

```
data(tank_comm)

# What is the largest possible C that I can use to calculate beta_C
calc_C_target(tank_comm)
```

---

calc_div	<i>Compute various diversity indices from a vector of species abundances (i.e., one row of a community matrix)</i>
----------	--

---

### Description

Compute various diversity indices from a vector of species abundances (i.e., one row of a community matrix)

**Usage**

```
calc_div(
  x,
  index,
  effort = NA,
  rare_thres = 0.05,
  PIE_replace = FALSE,
  C_target = NULL,
  extrapolate = TRUE,
  ...
)
```

**Arguments**

x	is a vector of species abundances
index	The calculated biodiversity indices. The options are <ul style="list-style-type: none"> <li>• N ... Number of individuals (total abundance)</li> <li>• S ... Number of species</li> <li>• S_n ... Rarefied or extrapolated number of species for n individuals</li> <li>• S_C ... Estimate species richness of a given level of coverage by C_target_gamma</li> <li>• S_asymp ... Estimated asymptotic species richness</li> <li>• f_0 ... Estimated number of undetected species</li> <li>• pct_rare ... The percent of rare species as defined by rare_thres</li> <li>• PIE ... Hurlbert's PIE (Probability of Interspecific Encounter)</li> <li>• S_PIE ... Effective number of species based on PIE</li> </ul> <p>See <i>Details</i> for additional information on the biodiversity statistics.</p>
effort	The standardized number of individuals used for the calculation of rarefied species richness. This must be a single integer.
rare_thres	The threshold that determines how the metric pct_rare is computed. It can range from (0, 1] and defaults to 0.05 which specifies that any species with less than or equal to 5 abundance in a sample is considered rare. It can also be specified as "N/S" which results in using average abundance as the threshold which McGill (2011) found to have the best small sample behavior.
PIE_replace	Used for PIE and SPIE. If TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).
C_target	When computing coverage based richness (S_C) then this argument can be used to specify the coverage to be used for the richness estimate. This defaults to NA in which case the target cover is computed by <code>calc_C_target</code> (i.e., the largest allowable sample size).
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method (Chao 1984, 1987). Defaults to TRUE.
...	additional arguments that can be passed to the function rarefaction when computing S_n.

**Examples**

```
data(inv_tank)
calc_div(tank_comm[1, ], 'S_n', effort = c(5, 10))
calc_div(tank_comm[1, ], 'S_C', C_target = 0.9)
```

calc\_PIE

*Calculate probability of interspecific encounter (PIE)***Description**

calc\_PIE returns the probability of interspecific encounter (PIE) which is also known as Simpson's evenness index and Gini-Simpson index.

**Usage**

```
calc_PIE(x, PIE_replace = FALSE)
```

**Arguments**

x	can either be a: 1) mob_in object, 2) community matrix-like object in which rows represent plots and columns represent species, or 3) a vector which contains the abundance of each species.
PIE_replace	if TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).

**Details**

By default, Hurlbert's (1971) sample-size corrected formula is used:

$$PIE = N / (N - 1) * (1 - \sum(p_i^2))$$

where N is the total number of individuals and  $p_i$  is the relative abundance of species i. This formulation uses sampling without replacement (replace = FALSE) For sampling with replacement (i.e., the sample-size uncorrected version), set replace = TRUE.

In earlier versions of mobr, there was an additional argument (ENS) for the conversion into an effective number of species (i.e S\_PIE). Now, calc\_SPIE has become its own function and the (ENS) argument is no longer supported. Please, use calc\_SPIE instead.

**Value**

either a single PIE value or vector of PIE values.

**Author(s)**

Dan McGlinn, Thore Engel

**References**

Hurlbert, S. H. (1971) The nonconcept of species diversity: a critique and alternative parameters. *Ecology* 52, 577-586.

**See Also**

[calc\\_SPIE](#)

**Examples**

```
data(inv_comm)
calc_PIE(inv_comm)
calc_PIE(inv_comm, PIE_replace = TRUE)
calc_PIE(c(23,21,12,5,1,2,3))
calc_PIE(c(23,21,12,5,1,2,3), PIE_replace = TRUE)
```

---

calc\_SPIE

*Calculate S\_PIE*

---

**Description**

S\_PIE is the effective number of species transformation of the probability of interspecific encounter (PIE) which is equal to the number of equally common species that result in that value of PIE.

**Usage**

```
calc_SPIE(x, PIE_replace = FALSE)
```

**Arguments**

x	can either be a: 1) mob_in object, 2) community matrix-like object in which rows represent plots and columns represent species, or 3) a vector which contains the abundance of each species.
PIE_replace	if TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).

**Details**

By default the sample size corrected version is returned (PIE\_replace = FALSE), which is the asymptotic estimator for the Hill number of diversity order q=2 (Chao et al, 2014). If PIE\_replace = TRUE the uncorrected hill number is returned. This is the same as `vegan::diversity(x, index="invsimpson")`.

**Value**

either a single S\_PIE value or vector of S\_PIE values.



## References

Chao, A., Gotelli, N. J., Hsieh, T. C., Sander, E. L., Ma, K. H., Colwell, R. K., & Ellison, A. M. (2014). Rarefaction and extrapolation with Hill numbers: A framework for sampling and estimation in species diversity studies. *Ecological Monographs* 84(1), 45-67.

## See Also

[calc\\_PIE](#)

## Examples

```
data(inv_comm)
calc_SPIE(inv_comm)
calc_SPIE(inv_comm, PIE_replace = TRUE)
calc_SPIE(c(23,21,12,5,1,2,3), PIE_replace=TRUE)
```

---

calc\_S\_C

*Calculate species richness for a given coverage level.*

---

## Description

This function uses coverage-based rarefaction to compute species richness. Specifically, the metric is computed as the

## Usage

```
calc_S_C(x, C_target = NULL, extrapolate = TRUE, interrupt = TRUE)
```

## Arguments

x	a site by species matrix or a species abundance distribution
C_target	target coverage between 0 and 1 (default is NULL). If not provided then target coverage is computed by <a href="#">calc_C_target</a>
extrapolate	logical. Defaults to TRUE in which case richness is extrapolated to sample sizes larger than observed in the dataset using the Chao1 method (Chao 1984, 1987).
interrupt	logical. Should the function throw an error when C_target exceeds the maximum recommendable coverage?

## Value

numeric value which is the species richness at a specific level of coverage.

## References

- Chao, A. 1984. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics* 11:265–270.
- Chao, A. 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Chao, A., and L. Jost. 2012. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. *Ecology* 93:2533–2547.
- Chao, A., N.J. Gotelli, T.C. Hsieh, E.L. Sander, K.H. Ma, R.K. Colwell, and A.M. Ellison 2014. Rarefaction and extrapolation with Hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological Monographs* 84:45-67.
- T. C. Hsieh, K. H. Ma and Anne Chao. 2024. iNEXT: iNterpolation and EXTrapolation for species diversity. R package version 3.0.1 URL: <http://chao.stat.nthu.edu.tw/wordpress/software-download/>.

## See Also

[invChat](#)

## Examples

```
data(tank_comm)
# What is species richness for a coverage value of 60%?
calc_S_C(tank_comm, C_target = 0.6)
```

---

Chat

*Calculate expected sample coverage  $C_{\hat{}}$*

---

## Description

Returns expected sample coverage of a sample ‘x’ for a smaller than observed sample size ‘m’ (Chao & Jost, 2012). This code was copied from iNEXT’s internal function `iNEXT::Chat.Ind` (Hsieh et al 2016).

## Usage

```
Chat(x, m)
```

## Arguments

x integer vector (species abundances)

m integer a number of individuals that is smaller than observed total community abundance.

## Value

a numeric value that is the expected coverage.

## References

Chao, A., and L. Jost. 2012. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. *Ecology* 93:2533–2547.

Anne Chao, Nicholas J. Gotelli, T. C. Hsieh, Elizabeth L. Sander, K. H. Ma, Robert K. Colwell, and Aaron M. Ellison 2014. Rarefaction and extrapolation with Hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological Monographs* 84:45-67.

T. C. Hsieh, K. H. Ma and Anne Chao. 2024. iNEXT: iNterpolation and EXTrapolation for species diversity. R package version 3.0.1 URL: <http://chao.stat.nthu.edu.tw/wordpress/software-download/>.

## Examples

```
data(inv_comm)
# What is the expected coverage at a sample size of 50 at the gamma scale?
Chat(colSums(inv_comm), 50)
```

---

```
compare_samp_rarefaction
```

*Compare all sample-based curves (random, spatially constrained-k-NN, spatially constrained-k-NCN)*

---

## Description

This is just plotting all curves.

## Usage

```
compare_samp_rarefaction(x)
```

## Arguments

x                    a mob\_in object

## Value

a plot

## Examples

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
compare_samp_rarefaction(inv_mob_in)
```

---

 fire\_comm

*Fire data set*


---

### Description

Woody plant species counts in burned and unburned forest sites in the Missouri Ozarks, USA.

### Details

fire\_comm is a site-by-species matrix with individual counts.

fire\_plot\_attr is a data frame with corresponding site variables. The column group specifies whether a site is "burned" or "unburned". This variable is considered a "treatment" in the mob framework. The columns x and y contain the spatial coordinates of the sites.

The data were adapted from Myers et al (2015).

### References

Myers, J. A., Chase, J. M., Crandall, R. M., & Jimenez, I. (2015). Disturbance alters beta-diversity but not the relative importance of community assembly mechanisms. *Journal of Ecology*, 103: 1291-1299.

### Examples

```
data(fire_comm)
data(fire_plot_attr)
fire_mob_in = make_mob_in(fire_comm, fire_plot_attr)
```

---

 get\_delta\_stats

*Conduct the MoB tests on drivers of biodiversity across scales.*


---

### Description

There are three tests, on effects of 1. the shape of the SAD, 2. treatment/group-level density, 3. degree of aggregation. The user can specifically to conduct one or more of these tests.

### Usage

```
get_delta_stats(
  mob_in,
  env_var,
  group_var = NULL,
  ref_level = NULL,
  tests = c("SAD", "N", "agg"),
  spat_algo = NULL,
  type = c("continuous", "discrete"),
```

```

stats = NULL,
inds = NULL,
log_scale = FALSE,
min_plots = NULL,
density_stat = c("mean", "max", "min"),
n_perm = 1000,
overall_p = FALSE
)

```

### Arguments

mob_in	an object of class mob_in created by make_mob_in()
env_var	a character string specifying the environmental variable in mob_in\$env to be used for explaining the change in richness
group_var	an optional character string in mob_in\$env which defines how samples are pooled. If not provided then each unique value of the argument env_var is used define the groups.
ref_level	a character string used to define the reference level of env_var to which all other groups are compared with. Only makes sense if env_var is a factor (i.e. type == 'discrete')
tests	specifies which one or more of the three tests ('SAD', 'N', 'agg') are to be performed. Default is to include all three tests.
spat_algo	character string that can be either: 'kNN' or 'kNCN' for k-nearest neighbor and k-nearest centroid neighbor sampling respectively. It defaults to k-nearest neighbor which is a more computationally efficient algorithm that closely approximates the potentially more correct k-NCN algo (see Details of ?rarefaction).
type	"discrete" or "continuous". If "discrete", pair-wise comparisons are conducted between all other groups and the reference group. If "continuous", a correlation analysis is conducted between the response variables and env_var.
stats	a vector of character strings that specifies what statistics to summarize effect sizes with. Options include: c('betas', 'r2', 'r2adj', 'f', 'p') for the beta-coefficients, r-squared, adjusted r-squared, F-statistic, and p-value respectively. The default value of NULL will result in only betas being calculated when type == 'discrete' and all possible stats being computed when type == 'continuous'. Note that for a discrete analysis all non-betas stats are meaningless because the model has zero degrees of freedom in this context.
inds	effort size at which the individual-based rarefaction curves are to be evaluated, and to which the sample-based rarefaction curves are to be interpolated. It can take three types of values, a single integer, a vector of integers, and NULL. If inds = NULL (the default), the curves are evaluated at every possible effort size, from 1 to the total number of individuals within the group (slow). If inds is a single integer, it is taken as the number of points at which the curves are evaluated; the positions of the points are determined by the "log_scale" argument. If inds is a vector of integers, it is taken as the exact points at which the curves are evaluated.

log_scale	if "inds" is given a single integer, "log_scale" determines the position of the points. If log_scale is TRUE, the points are equally spaced on logarithmic scale. If it is FALSE (default), the points are equally spaced on arithmetic scale.
min_plots	minimal number of plots for test 'agg', where plots are randomized within groups as null test. If it is given a value, all groups with fewer plots than min_plot are removed for this test. If it is NULL (default), all groups are kept. Warnings are issued if 1. there is only one group left and "type" is discrete, or 2. there are less than three groups left and "type" is continuous, or 3. reference group ("ref_group") is removed and "type" is discrete. In these three scenarios, the function will terminate. A different warning is issued if any of the remaining groups have less than five plots (which have less than 120 permutations), but the test will be carried out.
density_stat	reference density used in converting number of plots to numbers of individuals, a step in test "N". It can take one of the three values: "mean", "max", or "min". If it is "mean", the average plot-level abundance across plots (all plots when "type" is "continuous, all plots within the two groups for each pair-wise comparison when "type" is "discrete") are used. If it is "min" or "max", the minimum/maximum plot-level density is used.
n_perm	number of iterations to run for null tests, defaults to 1000.
overall_p	Boolean defaults to FALSE specifies if overall across scale p-values for the null tests. This should be interpreted with caution because the overall p-values depend on scales of measurement yet do not explicitly reflect significance at any particular scale.

**Value**

a "mob\_out" object with attributes

**Author(s)**

Dan McGlenn and Xiao Xiao

**See Also**

[rarefaction](#)

**Examples**

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
inv_mob_out = get_delta_stats(inv_mob_in, 'group', ref_level='uninvaded',
                             type='discrete', log_scale=TRUE, n_perm=3)
plot(inv_mob_out)
```

---

get_mob_stats	<i>Carry out biodiversity metric comparisons between groups.</i>
---------------	--

---

### Description

This function can compute a range of biodiversity metrics, their uncertainty, and carries out a permutation test to examine if groups differ in a particular metric more than would be expected due to random chance.

### Usage

```
get_mob_stats(
  mob_in,
  group_var,
  index = c("N", "S", "S_n", "S_PIE"),
  effort_samples = NULL,
  effort_min = 5,
  extrapolate = TRUE,
  return_NA = FALSE,
  rare_thres = 0.05,
  scales = c("alpha", "gamma", "beta"),
  PIE_replace = FALSE,
  C_target_gamma = NA,
  n_perm = 199,
  cl = NULL,
  ci = TRUE,
  ci_cent_stat = "median",
  ci_algo = "boot",
  ci_n_boot = 1000,
  ...
)
```

### Arguments

mob_in	an object of class mob_in created by make_mob_in()
group_var	String that specifies which variable in mob_in\$env the data should be grouped by.
index	The calculated biodiversity indices. The options are <ul style="list-style-type: none"> <li>• N ... Number of individuals (total abundance)</li> <li>• S ... Number of species</li> <li>• S_n ... Rarefied or extrapolated number of species for n individuals</li> <li>• S_C ... Estimate species richness of a given level of coverage by C_target_gamma</li> <li>• S_asymp ... Estimated asymptotic species richness</li> <li>• f_0 ... Estimated number of undetected species</li> <li>• pct_rare ... The percent of rare species as defined by rare_thres</li> </ul>

- PIE ... Hurlbert's PIE (Probability of Interspecific Encounter)
- S\_PIE ... Effective number of species based on PIE

See *Details* for additional information on the biodiversity statistics.

effort_samples	An integer that specifies the standardized number of individuals used for the calculation of rarefied species richness at the alpha-scale. It must be a single integer. The default value of NULL will result in the using the minimum number of individuals found across the samples is used, when this is not smaller than effort_min.
effort_min	The minimum number of individuals considered for the calculation of rarefied richness (Default value of 5). Samples with less individuals than effort_min are excluded from the analysis with a warning. Accordingly, when effort_samples is set by the user it has to be higher than effort_min.
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method (Chao 1984, 1987). Defaults to TRUE.
return_NA	Boolean in which the rarefaction function returns the observed S when effort is larger than the number of individuals. If set to TRUE then NA is returned. Note that this argument is only relevant when extrapolate = FALSE.
rare_thres	The threshold that determines how the metric pct_rare is computed. It can range from (0, 1] and defaults to 0.05 which specifies that any species with less than or equal to 5 abundance in a sample is considered rare. It can also be specified as "N/S" which results in using average abundance as the threshold which McGill (2011) found to have the best small sample behavior.
scales	The scales to compute the diversity indices for: <ul style="list-style-type: none"> <li>• alpha ... for each row of the site x species community matrix</li> <li>• gamma ... for the entire site x species community matrix</li> <li>• beta ... the ratio of diversity at the gamma and alpha scales.</li> </ul> Defaults to all three scales: c('alpha', 'gamma', 'beta')
PIE_replace	Used for PIE and SPIE. If TRUE, sampling with replacement is used. Otherwise, sampling without replacement (default).
C_target_gamma	When computing coverage based richness (S_C) then this argument can be used to specify the coverage to be used for the gamma scale richness estimate. This defaults to NA in which case the target cover is computed by <a href="#">calc_C_target</a> (i.e., the largest allowable sample size).
n_perm	number of iterations to run for null tests, defaults to 1000.
cl	A cluster object created by <a href="#">makeCluster</a> , or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations (see <i>Details</i> on performance). It can also be "future" to use a future backend (see <i>Details</i> ), NULL (default) refers to sequential evaluation.
ci	boolean, if TRUE then confidence intervals are calculated. Defaults to TRUE.
ci_cent_stat	a string that is either 'mean' or 'median' which specifies the measure of central tendency. Defaults to 'median'.
ci_algo	can be either 'boot' or 'loo' for bootstrap or leave-one-out methods respectively. Default value is 'boot'.
...	additional arguments that can be passed to <a href="#">calc_div</a>



## Details

This function is partially a wrapper for the functions `calc_comm_div` or `calc_comm_div_ci` that makes group comparisons easier to implement.

See `calc_comm_div` for more details on the biodiversity indices.

### Group comparison metric and test

For each metric group comparison the function computes  $D_{\text{bar}}$ : the average absolute difference between the groups. At the alpha scale the indices are averaged first before computing  $D_{\text{bar}}$ .

Permutation tests are carried out for testing differences of the biodiversity statistics among the groups (Legendre & Legendre 1998). This is accomplished by using  $D_{\text{bar}}$  as the test statistic and random shuffling the group label across the samples. The p-value indicates how many of the permutations result in a  $D_{\text{bar}}$  as large as the observed  $D_{\text{bar}}$  value.

## Value

A list of class `mob_stats` that contains two objects: 1) `comm_div` a data.frame of each diversity metrics for each group at each scale specified, and 2) `gorup_tests` a data.frame of the average difference between groups in their diversity metric ( $D_{\text{bar}}$ ) with an associated p-value derived from the permutation test.

## References

- Chao, A. 1984. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics* 11:265–270.
- Chao, A. 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Gotelli, N. J., and R. K. Colwell. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters* 4:379–391.
- Hurlbert, S. H. 1971. The nonconcept of species diversity: a critique and alternative parameters. *Ecology* 52:577–586.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88:2427–2439.
- McGill, B. J. 2011. Species abundance distributions. Pages 105-122 *Biological Diversity: Frontiers in Measurement and Assessment*, eds. A.E. Magurran and B.J. McGill.
- Whittaker, R. H. 1972. Evolution and measurement of species diversity. *Taxon* 21:213–251.

## Examples

```
# tank community analysis
data(tank_comm)
data(tank_plot_attr)
tank_mob <- make_mob_in(tank_comm, tank_plot_attr)
tank_stats <- get_mob_stats(tank_mob, 'group', index = c('S', 'S_PIE', 'S_C'),
                           n_perm = 19)

tank_stats
```

---

get_null_comm	<i>Generate a null community matrix</i>
---------------	---

---

### Description

Three null models are implemented that randomize different components of community structure while keeping other components constant.

### Usage

```
get_null_comm(comm, null_model, groups = NULL)
```

### Arguments

comm	community matrix of abundances with plots as rows and species columns.
null_model	a string which specifies which null model to use options include: 'rand_SAD', 'rand_N', and 'rand_agg'. See Details for description of each null model.
groups	optional argument that is a vector of group ids which specify which group each site is associated with. If is NULL then all rows of the community matrix are assumed to be members of the same group

### Details

This function implements three different nested null models. They are considered nested because at the core of each null model is the random sampling with replacement of the relative abundance distribution (RAD) to generate a random sample of a species abundance distribution (SAD). Here we describe each null model:

- 'rand\_SAD' ... A random SAD is generated using a sample with replacement of individuals from the species pool proportional to their observed relative abundance. This null model will produce an SAD that is of a similar functional form to the observed SAD (Green and Plotkin 2007). The total abundance of the random SAD is the same as the observed SAD but overall species richness will be equal to or less than the observed SAD. This algorithm ignores the group argument. This sampling algorithm is also used in the two other null models 'rand\_N' and 'rand\_agg'.
- 'rand\_N' ... The total number of individuals in a plot is shuffled across all plots (within and between groups). Then for each plot that many individuals are drawn randomly from the group specific relative abundance distribution with replacement for each plot (i.e., using the 'rand\_SAD' algorithm described above. This removes group differences in the total number of individuals in a given plot, but maintains group level differences in their SADs.
- 'rand\_agg' ... This null model nullifies the spatial structure of individuals (i.e., their aggregation), but it is constrained by the observed total number of individuals in each plot (in contrast to the 'rand\_N' null model), and the group specific SAD (in contrast to the 'rand\_SAD' null model). The other two null models also nullify spatial structure. The 'rand\_agg' null model is identical to the 'rand\_N' null model except that plot abundances are not shuffled.

Replaces depreciated function 'permute\_comm'

**Value**

a site-by-species matrix

**References**

Green, J. L., and J. B. Plotkin. 2007. A statistical theory for sampling species abundances. *Ecology Letters* 10:1037-1045.

**Examples**

```
S = 3
N = 20
nplots = 4
comm = matrix(rpois(S * nplots, 1), ncol = S, nrow = nplots)
comm
groups = rep(1:2, each=2)
groups
set.seed(1)
get_null_comm(comm, 'rand_SAD')
# null model 'rand_SAD' ignores groups argument
set.seed(1)
get_null_comm(comm, 'rand_SAD', groups)
set.seed(1)
get_null_comm(comm, 'rand_N')
# null model 'rand_N' does not ignore the groups argument
set.seed(1)
get_null_comm(comm, 'rand_N', groups)
# note that the 'rand_agg' null model is constrained by observed plot abundances
noagg = get_null_comm(comm, 'rand_agg', groups)
noagg
rowSums(comm)
rowSums(noagg)
```

---

get_samples	<i>Generate distribution of sampled community matrices from an original matrix.</i>
-------------	---

---

**Description**

The sampled matrices are either bootstrap or leave-one-out (the default) samples. The bootstrap samples represent a random set of the rows sampled with replacement from the original matrix. The leave-one-out samples are generated by removing each sample one at a time from the original community matrix.

**Usage**

```
get_samples(abund_mat, algo = "loo", n_boot = 1000)
```

**Arguments**

abund_mat	Abundance based site-by-species table. Species as columns
algo	can be either 'boot' or 'loo' for bootstrap or leave-one-out methods respectively. Default value is 'loo'.
n_boot	how to many boot strapped samples to create, defaults to 1000.

**Details**

These sampled community matrices can become the input to `calc_comm_div_ci` for computing confidence intervals of diversity metrics.

Note, it is unclear which sampling algorithm (bootstrap or loo) is least biased and most efficient for the purpose of generating confidence intervals.

**Value**

a list of community matrices which are sampled from the original input matrix.

**Author(s)**

Felix May and Dan McGlenn

**References**

- Chao, A. 1984. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics* 11:265–270.
- Chao, A. 1987. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Gotelli, N. J., and R. K. Colwell. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters* 4:379–391.
- Hurlbert, S. H. 1971. The nonconcept of species diversity: a critique and alternative parameters. *Ecology* 52:577–586.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88:2427–2439.
- McGill, B. J. 2011. Species abundance distributions. Pages 105-122 *Biological Diversity: Frontiers in Measurement and Assessment*, eds. A.E. Magurran and B.J. McGill.
- Whittaker, R. H. 1972. Evolution and measurement of species diversity. *Taxon* 21:213–251.

**See Also**

[calc\\_comm\\_div\\_ci](#)

**Examples**

```
data(tank_comm)
# 2 leave-one-out samples
lapply(get_samples(tank_comm)[1:2], head)
# 2 bootstrap samples
lapply(get_samples(tank_comm, algo = 'boot', n_boot = 2), head)
```

---

invChat	<i>Number of individuals corresponding to a desired coverage (inverse <math>C_{\hat{}}</math>)</i>
---------	--

---

### Description

If you wanted to resample a vector to a certain expected sample coverage, how many individuals would you have to draw? This is  $C_{\hat{}}$  solved for the number of individuals. This code is a modification of INEXT's internal function 'invChat.Ind' (Hsieh et al 2016).

### Usage

```
invChat(x, C)
```

### Arguments

x	integer vector (species abundances)
C	coverage value between 0 and 1

### Value

a numeric value which is the number of individuals for a given level of coverage C.

### References

Chao, A., and L. Jost. 2012. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. *Ecology* 93:2533–2547.

Anne Chao, Nicholas J. Gotelli, T. C. Hsieh, Elizabeth L. Sander, K. H. Ma, Robert K. Colwell, and Aaron M. Ellison 2014. Rarefaction and extrapolation with Hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological Monographs* 84:45-67.

T. C. Hsieh, K. H. Ma and Anne Chao. 2024. iNEXT: iNterpolation and EXTrapolation for species diversity. R package version 3.0.1 URL: <http://chao.stat.nthu.edu.tw/wordpress/software-download/>.

### See Also

[calc\\_S\\_C](#)

### Examples

```
data(inv_comm)
# What sample size corresponds to an expected sample coverage of 55%?
invChat(colSums(inv_comm), 0.55)
```

---

inv_comm	<i>Invasive plants dataset</i>
----------	--------------------------------

---

### Description

Herbaceous plant species counts sites invaded and uninvaded by *Lonicera maackii* (Amur honeysuckle) which is an invasive shrub.

### Details

inv\_comm is a site-by-species matrix with individual counts.

inv\_plot\_attr is a data frame with corresponding site variables. The column group specifies whether a site is "invaded" or "uninvaded". This variable is considered a "treatment" in the mob framework. The columns x and y contain the spatial coordinates of the sites.

The data were adapted from Powell et al (2013).

### References

Powell, K. I., Chase, J. M., & Knight, T. M. (2013). Invasive plants have scale-dependent effects on diversity by altering species-area relationships. *Science*, 339: 316-318.

### Examples

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr)
```

---

kNCN_average	<i>Construct spatially constrained sample-based rarefaction (sSBR) curve using the k-Nearest-Centroid-neighbor (k-NCN) algorithm</i>
--------------	--

---

### Description

This function accumulates samples according their proximity to all previously included samples (their centroid) as opposed to the proximity to the initial focal sample. This ensures that included samples mutually close to each other and not all over the place.

### Usage

```
kNCN_average(
  x,
  n = NULL,
  coords = NULL,
  repetitions = 1,
  no_pb = TRUE,
```

```

    latlong = FALSE,
    cl = NULL
  )

```

### Arguments

x	a mob_in object or a community site x species matrix
n	number of sites to include.
coords	spatial coordinates of the samples. If x is a mob_in object, the function uses its 'spat' table as coordinates.
repetitions	Number of times to repeat the procedure. Useful in situations where there are many ties in the distance matrix.
no_pb	binary, if TRUE then a progress bar is not printed, defaults to TRUE
latlong	if longitude latitudes are supplied
cl	A cluster object created by <a href="#">makeCluster</a> , or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations (see <a href="#">Details</a> on performance). It can also be "future" to use a future backend (see <a href="#">Details</a> ), NULL (default) refers to sequential evaluation.

### Details

Internally the function constructs one curve per sample whereby each sample serves as the initial sample repetition times. Finally, the average curve is returned.

### Value

a numeric vector of estimated species richness

### Examples

```

data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
kNCN_average(inv_mob_in, n = 5)

# parallel evaluation using the parallel package
# run in parallel
library(parallel)
cl = makeCluster(2L)
clusterEvalQ(cl, library(mobr))
clusterExport(cl, 'inv_mob_in')
S_kNCN = kNCN_average(inv_mob_in, cl=cl)

stopCluster(cl)

```

---

make\_mob\_in                      *Create the 'mob\_in' object.*

---

### Description

The 'mob\_in' object will be passed on for analyses of biodiversity across scales.

### Usage

```
make_mob_in(  
  comm,  
  plot_attr,  
  coord_names = NULL,  
  binary = FALSE,  
  latlong = FALSE  
)
```

### Arguments

comm	community matrix in which rows are samples (e.g., plots) and columns are species.
plot_attr	matrix which includes the environmental attributes and spatial coordinates of the plots. Environmental attributes are mandatory, while spatial coordinates are optional.
coord_names	character vector with the names of the columns of plot_attr that specify the coordinates of the samples. Defaults to NULL (no coordinates). When providing coordinate names, the order the names are provided matters when working with latitude-longitude coordinates (i.e., argument latlong = TRUE, and it is expected that the column specifying the x-coordinate or the longitude is provided first, y-coordinate or latitude provided second. To provide coordinate names use the following syntax: coord_names = c('longitude_col_name', 'latitude_col_name')
binary	Boolean, defaults to FALSE. Whether the plot by species matrix "comm" is in abundances or presence/absence.
latlong	Boolean, defaults to FALSE. Whether the coordinates are latitude-longitudes. If TRUE, distance calculations by downstream functions are based upon great circle distances rather than Euclidean distances. Note latitude-longitudes should be in decimal degree.

### Value

a "mob\_in" object with four attributes. "comm" is the plot by species matrix. "env" is the environmental attribute matrix, without the spatial coordinates. "spat" contains the spatial coordinates (1-D or 2-D). "tests" specifies whether each of the three tests in the biodiversity analyses is allowed by data.



**Author(s)**

Dan McGlinn and Xiao Xiao

**Examples**

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
```

---

mobr

*Measurement of Biodiversity in R*

---

**Description**

The primary aim of this package is to provide ecologist's tools to examine changes in biodiversity across spatial scales. Additionally, the package provides a method to examine how a factor mediates species richness via its effects on different aspects of community structure: total abundance, species commonness, and spatial aggregation of conspecifics.

**Author(s)**

**Maintainer:** Daniel McGlinn <danmcglinn@gmail.com> [copyright holder]

Authors:

- Xiao Xiao <xiao@weecology.org>
- Brian McGill <brimcgill@gmail.com>
- Felix May <felix.may@posteo.de>
- Thore Engel <thore.engel@idiv.de>
- Caroline Oliver <olivercs@g.cofc.edu>
- Shane Blowes <shane.blowes@idiv.de>
- Tiffany Knight <tiffany.knight@idiv.de>
- Oliver Purschke <oliverpurschke@web.de>
- Nicholas Gotelli <Nicholas.Gotelli@uvm.edu>
- Jon Chase <jonathan.chase@idiv.de>

**See Also**

Useful links:

- <https://github.com/MoBiodiv/mobr>
- Report bugs at <https://github.com/MoBiodiv/mobr/issues>

---

plot.mob_out	<i>Plot the multiscale MoB analysis output generated by get_delta_stats.</i>
--------------	--

---

### Description

Plot the multiscale MoB analysis output generated by `get_delta_stats`.

### Usage

```
## S3 method for class 'mob_out'
plot(
  x,
  stat = "b1",
  log2 = "",
  scale_by = NULL,
  display = c("S ~ effort", "effect ~ grad", "stat ~ effort"),
  eff_sub_effort = TRUE,
  eff_log_base = 2,
  eff_disp_pts = TRUE,
  eff_disp_smooth = FALSE,
  ...
)
```

### Arguments

x	a <code>mob_out</code> class object
stat	a character string that specifies what statistic should be used in the effect size plots. Options include: <code>c('b0', 'b1', 'r', 'r2', 'r2adj', 'f')</code> for the beta-coefficients, person correlation coefficient, r-squared, adjusted r-squared, and F-statistic respectively. If the explanatory variable is a factor then 'b1' is the only reasonable option. The default is set to the regression slope 'b1' because this appears to have the strongest statistical power.
log2	a character string specifying if the x- or y-axis should be rescale by log base 2. Only applies when <code>display == 'S ~ effort'   'S ~ effort'</code> . Options include: <code>c('', 'x', 'y', 'xy')</code> for no rescaling, x-axis, y-axis, and both x and y-axes respectively. Default is set to no rescaling.
scale_by	a character string specifying if sampling effort should be rescaled. Options include: <code>NULL</code> , 'indiv', and 'plot' for no rescaling, rescaling to number of individuals, and rescaling to number of plots respectively. The rescaling is carried out using <code>mob_out\$density_stat</code> .
display	a string that specifies what graphical panels to display. Options include: <ul style="list-style-type: none"> <li>• <code>S ~ expl ...</code> plot of S versus the explanatory variable</li> <li>• <code>S ~ effort ...</code> plot of S versus sampling effort (i.e., a rarefaction curve)</li> </ul>

- effect ~ expl ... plot of agg., N, and SAD effect size versus explanatory variable
- stat ~ effort ... plot of summary statistic versus sampling effort

Defaults to 'S ~ effort', 'effect ~ expl', and 'stat ~ effort'.

eff_sub_effort	Boolean which determines if only a subset of efforts will be considered in the plot of effect size (i.e., when display = 'effect ~ expl'. Defaults to TRUE to declutter the plots.
eff_log_base	a positive real number that determines the base of the logarithm that efforts were be distributed across, the larger this number the fewer efforts will be displayed.
eff_disp_pts	Boolean to display the raw effect points, defaults to TRUE
eff_disp_smooth	Boolean to display the regressions used to summarize the linear effect of the explanatory variable on the effect sizes, defaults to FALSE
...	parameters passed to other functions

**Value**

plots the effect of the SAD, the number of individuals, and spatial aggregation on the difference in species richness

**Author(s)**

Dan McGlenn and Xiao Xiao

**Examples**

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
inv_mob_out = get_delta_stats(inv_mob_in, 'group', ref_level='uninvaded',
                             type='discrete', log_scale=TRUE, n_perm=4)
plot(inv_mob_out, 'b1')

plot(inv_mob_out, 'b1', scale_by = 'indiv')
```

---

plot.mob\_stats                      *Plot statistics for a MoB analysis*

---

**Description**

Plots a mob\_stats object which is produced by the function `get_mob_stats`. The p-value for each statistic is displayed in the plot title if applicable.

**Usage**

```
## S3 method for class 'mob_stats'
plot(mob_stats, group_var, group_order = NULL)
```

**Arguments**

mob_stats	a mob_stats object produced by the function <a href="#">get_mob_stats</a>
group_var	a character string which specifies which variable provides the groups that the diversity indices are compared across.
group_order	Optional vector of group levels the order of which defines the order in which the groups are plotted from left-to-right. Note that the default in R is to order groups alphabetically.

**Examples**

```

data(tank_comm)
data(tank_plot_attr)
tank_mob <- make_mob_in(tank_comm, tank_plot_attr)
# for quick results we specify a low number of permutations
# and bootstrap samples these should be increased in real
# analyses.
tank_stats <- get_mob_stats(tank_mob, 'group',
                           index = c('N', 'S', 'S_n', 'S_PIE', 'S_C'),
                           n_perm = 19, ci_n_boot = 20)
p <- plot(tank_stats, 'group')
# plot of S
p$S
# change the order of the groups on the plot
p <- plot(tank_stats, 'group', group_order = c('low', 'high'))
p$S

```

---

plot\_abu

*Plot distributions of species abundance*


---

**Description**

Plot distributions of species abundance

**Usage**

```

plot_abu(
  mob_in,
  group_var,
  ref_level = NULL,
  type = "sad",
  scale = "gamma",
  col = NULL,
  lwd = 3,
  log = "",
  leg_loc = "topleft"
)

```

**Arguments**

mob_in	a 'mob_in' class object produced by 'make_mob_in'
group_var	String that specifies which field in mob_in\$env the data should be grouped by
ref_level	String that defines the reference level of group_var to which all other groups are compared with, defaults to NULL. If NULL then the default contrasts of group_var are used.
type	either 'sad' or 'rad' for species abundance vs rank abundance distribution, defaults to 'sad'.
scale	character string either 'alpha' for sample scale or 'gamma' for group scale. Defaults to 'gamma'.
col	optional vector of colors.
lwd	a number which specifies the width of the lines
log	a string that specifies if any axes are to be log transformed, options include 'x', 'y' or 'xy' in which either the x-axis, y-axis, or both axes are log transformed respectively
leg_loc	a string that specifies the location of the legend, options include: 'lowerleft', 'topleft', 'lowerright', 'topright'

**Examples**

```
data(inv_comm)
data(inv_plot_attr)
inv_mob_in <- make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
plot_abu(inv_mob_in, 'group', 'uninvaded', type='sad', log='x')
plot_abu(inv_mob_in, 'group', 'uninvaded', type='rad', scale = 'alpha', log='x')
```

---

plot_comm_div	<i>Plot alpha-, beta-, and gamma-scale biodiversity statistics for a MoB analysis</i>
---------------	---

---

**Description**

Plots the community diversity metrics from produced by the function calc\_comm\_div. The p-value for each statistic is displayed in the plot title if applicable.

**Usage**

```
plot_comm_div(
  comm_div,
  index = NULL,
  multi_panel = FALSE,
  col = c("#FFB3B5", "#78D3EC", "#6BDABD", "#C5C0FE", "#E2C288", "#F7B0E6", "#AAD28C"),
  cex.axis = 1.2,
  ...
)
```

**Arguments**

comm_div	a table that is output by <code>calc_comm_div</code> that has the sample (alpha) and group (gamma) level statistics
index	The biodiversity statistics that should be plotted. See <code>calc_comm_div</code> for information on the indices. By default there is one figure for each index, with panels for alpha- and gamma-scale results as well as for beta-diversity when applicable.
multi_panel	A logical variable. If <code>multi_panel = TRUE</code> then a multipanel plot is produced, which shows observed, rarefied, and asymptotic species richness and S_PIE at the alpha- and gamma-scale. This set of variables conveys a comprehensive picture of the underlying biodiversity changes.
col	a vector of colors for the groups, set to NA if no color is preferred
cex.axis	The magnification to be used for axis annotation relative to the current setting of <code>cex</code> . Defaults to 1.2.
...	additional arguments to provide to <code>boxplot</code> , <code>points</code> , and confidence interval functions

**Details**

The user may specify which results to plot or simply to plot all the results.

**Author(s)**

Felix May, Xiao Xiao, and Dan McGlenn

**Examples**

```
library(dplyr)
data(tank_comm)
data(tank_plot_attr)
indices <- c('N', 'S', 'S_C', 'S_n', 'S_PIE')
# the grouping variable must be called group in the
# tank_div data.frame
tank_div <- tibble(tank_comm) |>
  group_by(group = tank_plot_attr$group) |>
  group_modify(~ calc_comm_div(.x, index = indices, effort = 5,
                              extrapolate = TRUE))

# plot the community metrics
plot_comm_div(tank_div, index = "S")
plot_comm_div(tank_div, index = "S_n")
# or plot all of the indices at once with
plot_comm_div(tank_div)
```

---

plot_N	<i>Plot the relationship between the number of plots and the number of individuals</i>
--------	--

---

### Description

The MoB methods assume a linear relationship between the number of plots and the number of individuals. This function provides a means of verifying the validity of this assumption

### Usage

```
plot_N(comm, n_perm = 1000)
```

### Arguments

comm	community matrix with sites as rows and species as columns
n_perm	number of permutations to average across, defaults to 1000

### Author(s)

Dan McGlenn

### Examples

```
data(inv_comm)
plot_N(inv_comm)
```

---

plot_rarefaction	<i>Plot rarefaction curves for each treatment group</i>
------------------	---

---

### Description

Plot rarefaction curves for each treatment group

### Usage

```
plot_rarefaction(
  mob_in,
  group_var,
  ref_level = NULL,
  method,
  spat_algo = NULL,
  dens_ratio = 1,
  scales = c("alpha", "gamma", "study"),
  raw = TRUE,
  smooth = FALSE,
```

```

    avg = FALSE,
    col = NULL,
    lwd = 3,
    log = "",
    leg_loc = "topleft",
    one_panel = FALSE,
    ...
)

```

### Arguments

<code>mob_in</code>	a 'mob_in' class object produced by 'make_mob_in'
<code>group_var</code>	String that specifies which field in <code>mob_in\$env</code> the data should be grouped by
<code>ref_level</code>	String that defines the reference level of <code>group_var</code> to which all other groups are compared with, defaults to NULL. If NULL then the default contrasts of <code>group_var</code> are used.
<code>method</code>	<p>a character string that specifies the method of rarefaction curve construction it can be one of the following:</p> <ul style="list-style-type: none"> <li>• 'IBR' ... individual-based rarefaction in which species are accumulated by randomly sampling individuals</li> <li>• 'SBR' ... sample-based rarefaction in which species are accumulated by randomly sampling samples (i.e., plots). Note that within plot spatial aggregation is maintained with this approach. Although this curve is implemented here, it is not used in the current version of the MoB framework</li> <li>• 'nsSBR' ... non-spatial, sampled-based rarefaction in which species are accumulated by randomly sampling samples that represent a spatially random sample of individuals (i.e., no with-in plot spatial aggregation). The argument <code>dens_ratio</code> must also be set otherwise this sampling results in a curve identical to the IBR (see Details).</li> <li>• 'sSBR' ... spatial sample-based rarefaction in which species are accumulated by including spatially proximate samples first.</li> <li>• 'spexSBR' ... spatially-explicit sample-based rarefaction in which species are accumulated as in 'sSBR' but sampling effort is not measured by no. of samples, but by cumulative distance or cumulative area as specified by 'spat_algo' (see details)</li> </ul>
<code>spat_algo</code>	character string that can be either: 'kNN', 'kNCN', or 'convexhull' for k-nearest neighbor, k-nearest centroid neighbor sampling, or convex-hull polygon calculation respectively. It defaults to k-nearest neighbor which is a more computationally efficient algorithm that closely approximates the potentially more correct k-NCN algo (see Details). Currently, 'kNN' and 'k-NCN' are available for method 'ssBR', while 'kNN' 'convexhull' are available for method 'spexSBR'.
<code>dens_ratio</code>	the ratio of individual density between a reference group and the community data (i.e., x) under consideration. This argument is used to rescale the rarefaction curve when estimating the effect of individual density on group differences in richness.



scales	character string which defaults to c('alpha', 'gamma', 'study') indicating that rarefaction curves at the alpha (i.e., single plot), gamma (i.e., group of plots), and study (i.e., all plots) scales should be computed and plotted.
raw	boolean. Defaults to TRUE so that raw rarefaction curves without averaging or smoothing are plotted
smooth	boolean. Defaults to FALSE. If set to TRUE a lowess smoother is used on the 'alpha' scale curves. Has no effect at gamma or study scales
avg	boolean. Defaults to FALSE. If set to TRUE then the average richness across the groups is computed and plotted.
col	optional vector of colors.
lwd	a number which specifies the width of the lines
log	a string that specifies if any axes are to be log transformed, options include 'x', 'y' or 'xy' in which either the x-axis, y-axis, or both axes are log transformed respectively
leg_loc	a string that specifies the location of the legend, options include: 'lowerleft', 'topleft', 'lowerright', 'topright'
one_panel	boolean. Defaults to FALSE. If set to TRUE then the alpha scale and gamma scale curves are put on the same graph.
...	other arguments to provide to <a href="#">rarefaction</a>

### Examples

```

data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
# random individual based rarefaction curves
par(mfrow=c(1,2))
plot_rarefaction(inv_mob_in, 'group', 'uninvaded', 'IBR',
                 leg_loc='bottomright')
plot_rarefaction(inv_mob_in, 'group', 'uninvaded', 'IBR',
                 log='xy')
# random sample based rarefaction curves
plot_rarefaction(inv_mob_in, 'group', 'uninvaded', 'SBR', log='xy',
                 leg_loc='bottomright')
# spatial sample based rarefaction curves
plot_rarefaction(inv_mob_in, 'group', 'uninvaded', 'sSBR', log='xy',
                 leg_loc='bottomright', avg = TRUE, smooth = TRUE)

```

---

rarefaction

*Rarefied Species Richness*

---

### Description

The expected number of species given a particular number of individuals or samples under random and spatially explicit nearest neighbor sampling

**Usage**

```
rarefaction(
  x,
  method,
  effort = NULL,
  coords = NULL,
  latlong = NULL,
  dens_ratio = 1,
  extrapolate = FALSE,
  return_NA = FALSE,
  quiet_mode = FALSE,
  spat_algo = NULL,
  sd = FALSE
)
```

**Arguments**

- |        |   |
|--------|---|
| x      | can either be a: 1) <code>mob_in</code> object, 2) community matrix-like object in which rows represent plots and columns represent species, or 3) a vector which contains the abundance of each species.   |
| method | <p>a character string that specifies the method of rarefaction curve construction it can be one of the following:</p> <ul style="list-style-type: none"> <li>• 'IBR' ... individual-based rarefaction in which species are accumulated by randomly sampling individuals</li> <li>• 'SBR' ... sample-based rarefaction in which species are accumulated by randomly sampling samples (i.e., plots). Note that within plot spatial aggregation is maintained with this approach. Although this curve is implemented here, it is not used in the current version of the MoB framework</li> <li>• 'nsSBR' ... non-spatial, sampled-based rarefaction in which species are accumulated by randomly sampling samples that represent a spatially random sample of individuals (i.e., no with-in plot spatial aggregation). The argument <code>dens_ratio</code> must also be set otherwise this sampling results in a curve identical to the IBR (see Details).</li> <li>• 'sSBR' ... spatial sample-based rarefaction in which species are accumulated by including spatially proximate samples first.</li> <li>• 'spexSBR' ... spatially-explicit sample-based rarefaction in which species are accumulated as in 'sSBR' but sampling effort is not measured by no. of samples, but by cumulative distance or cumulative area as specified by 'spat_algo' (see details)</li> </ul> |
| effort | optional argument to specify what number of individuals, number of samples, or spatial sampling effort (i.e., cumulative distance or area) depending on 'method' to compute rarefied richness as. If not specified all possible values from 1 to the maximum sampling effort are used   |
| coords | an optional matrix of geographic coordinates of the samples. Only required when using the spatial rarefaction method and this information is not already supplied by <code>x</code> . The first column should specify the x-coordinate (e.g., longitude) and the second coordinate should specify the y-coordinate (e.g., latitude)   |

latlong	Boolean if coordinates are latitude-longitude decimal degrees
dens_ratio	the ratio of individual density between a reference group and the community data (i.e., $x$ ) under consideration. This argument is used to rescale the rarefaction curve when estimating the effect of individual density on group differences in richness.
extrapolate	Boolean which specifies if richness should be extrapolated when effort is larger than the number of individuals using the chao1 method. Defaults to FALSE in which case it returns observed richness. Extrapolation is only implemented for individual-based rarefaction (i.e., <code>method = 'indiv'</code> )
return_NA	Boolean defaults to FALSE in which the function returns the observed $S$ when effort is larger than the number of individuals or number of samples (depending on the method of rarefaction). If set to TRUE then NA is returned. Note that this argument is only relevant when <code>extrapolate = FALSE</code> .
quiet_mode	Boolean defaults to FALSE, if TRUE then warnings and other non-error messages are suppressed.
spat_algo	character string that can be either: 'kNN', 'kNCN', or 'convexhull' for k-nearest neighbor, k-nearest centroid neighbor sampling, or convex-hull polygon calculation respectively. It defaults to k-nearest neighbor which is a more computationally efficient algorithm that closely approximates the potentially more correct k-NCN algo (see Details). Currently, 'kNN' and 'k-NCN' are available for method 'ssBR', while 'kNN' 'convexhull' are available for method 'spexSBR'.
sd	Boolean defaults to FALSE, if TRUE then standard deviation of richness is also returned using the formulation of Heck 1975 Eq. 2.

## Details

The analytical formulas of Cayuela et al. (2015) are used to compute the random sampling expectation for the individual and sampled based rarefaction methods. The spatially constrained rarefaction curve (Chiarucci et al. 2009) also known as the sample-based accumulation curve (Gotelli and Colwell 2001) can be computed in one of two ways which is determined by the argument `spat_algo`. In the kNN approach each plot is accumulated by the order of their spatial proximity to the original focal cell. If plots have the same distance from the focal plot then one is chosen randomly to be sampled first. In the kNCN approach, a new centroid is computed after each plot is accumulated, then distances are recomputed from that new centroid to all other plots and the next nearest is sampled. The kNN is faster because the distance matrix only needs to be computed once, but the sampling of kNCN which simultaneously minimizes spatial distance and extent is more similar to an actual person searching a field for species. For both kNN and kNCN, each plot in the community matrix is treated as a starting point and then the mean of these  $n$  possible accumulation curves is computed.

For individual-based rarefaction if effort is greater than the number of individuals and `extrapolate = TRUE` then the Chao1 method is used (Chao 1984, 1987). The code used to perform the extrapolation was ported from `iNext::D0.hat` found at <https://github.com/JohnsonHsieh/iNEXT>. T. C. Hsieh, K. H. Ma and Anne Chao are the original authors of the `iNEXT` package.

If effort is greater than sample size and `extrapolate = FALSE` then the observed number of species is returned.

Standard deviation of richness can only be computed for individual based rarefaction and it is assigned as an attribute (see examples). The code for this computation was ported from `vegan::rarefy` (Oksanen et al. 2022)

### Value

A vector of rarefied species richness values

### Author(s)

Dan McGlinn and Xiao Xiao

### References

- Cayuela, L., N.J. Gotelli, & R.K. Colwell (2015) Ecological and biogeographic null hypotheses for comparing rarefaction curves. *Ecological Monographs*, 85, 437-454. Appendix A: <http://esapubs.org/archive/mono/M085/01A.php>
- Chao, A. (1984) Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of Statistics*, 11, 265-270.
- Chao, A. (1987) Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43, 783-791.
- Chiarucci, A., G. Bacaro, D. Rocchini, C. Ricotta, M. Palmer, & S. Scheiner (2009) Spatially constrained rarefaction: incorporating the autocorrelated structure of biological communities into sample-based rarefaction. *Community Ecology*, 10, 209-214.
- Gotelli, N.J. & Colwell, R.K. (2001) Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters*, 4, 379-391.
- Heck, K.L., van Belle, G. & Simberloff, D. (1975). Explicit calculation of the rarefaction diversity measurement and the determination of sufficient sample size. *Ecology* 56, 1459–1461.
- Oksanen, J. et al. 2022. *Vegan: community ecology package*. R package version 2.6-4. <https://CRAN.R-project.org/package=vegan>

### Examples

```
data(inv_comm)
data(inv_plot_attr)
sad = colSums(inv_comm)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
# rarefaction can be performed on different data inputs
# all three give same answer
# 1) the raw community site-by-species matrix
rarefaction(inv_comm, method='IBR', effort=1:10)
# 2) the SAD of the community
rarefaction(inv_comm, method='IBR', effort=1:10)
# 3) a mob_in class object
# the standard deviation of the richness estimates for IBR may be returned
# which is helpful for computing confidence intervals
S_n <- rarefaction(inv_comm, method='IBR', effort=1:10, sd=TRUE)
attr(S_n, 'sd')
plot(1:10, S_n, ylim=c(0,8), type = 'n')
```

```

z <- qnorm(1 - 0.05 / 2)
hi <- S_n + z * attr(S_n, 'sd')
lo <- S_n - z * attr(S_n, 'sd')
attributes(hi) <- NULL
attributes(lo) <- NULL
polygon(c(1:10, 10:1), c(hi, rev(lo)), col='grey', border = NA)
lines(1:10, S_n, type = 'o')
# rescaling of individual based rarefaction
# when the density ratio is 1 the richness values are
# identical to the unscaled rarefaction
rarefaction(inv_comm, method='IBR', effort=1:10, dens_ratio=1)
# however the curve is either shrunk when density is higher than
# the reference value (i.e., dens_ratio < 1)
rarefaction(inv_comm, method='IBR', effort=1:10, dens_ratio=0.5)
# the curve is stretched when density is lower than the
# reference value (i.e., dens_ratio > 1)
rarefaction(inv_comm, method='IBR', effort=1:10, dens_ratio=1.5)
# sample based rarefaction under random sampling
rarefaction(inv_comm, method='SBR')

# sampled based rarefaction under spatially explicit nearest neighbor sampling
rarefaction(inv_comm, method='sSBR', coords=inv_plot_attr[, c('x','y')],
            latlong=FALSE)
# the syntax is simpler if supplying a mob_in object
rarefaction(inv_mob_in, method='sSBR', spat_algo = 'kNCN')
rarefaction(inv_mob_in, method='sSBR', spat_algo = 'kNN')
rarefaction(inv_mob_in, method='spexSBR', spat_algo = 'kNN')

```

---

subset.mob\_in

*Subset the rows of the mob data input object*


---

## Description

This function subsets the rows of comm, env, and spat attributes of the mob\_in object

## Usage

```

## S3 method for class 'mob_in'
subset(x, subset, type = "string", drop_levels = FALSE, ...)

```

## Arguments

x	an object of class mob_in created by <a href="#">make_mob_in</a>
subset	expression indicating elements or rows to keep: missing values are taken as false.
type	specifies the type of object the argument subset specifies, may be: string, integer, or logical, defaults to string
drop_levels	Boolean if TRUE unused levels are removed from factors in mob_in\$env
...	parameters passed to other functions

**Examples**

```

data(inv_comm)
data(inv_plot_attr)
inv_mob_in = make_mob_in(inv_comm, inv_plot_attr, coord_names = c('x', 'y'))
subset(inv_mob_in, group == 'invaded')
subset(inv_mob_in, 1:4, type='integer')
subset(inv_mob_in, 1:4, type='integer', drop_levels=TRUE)
sub_log = c(TRUE, FALSE, TRUE, rep(FALSE, nrow(inv_mob_in$comm) - 3))
subset(inv_mob_in, sub_log, type='logical')

```

---

tank\_comm

*Cattle tank data set*


---

**Description**

Species counts of aquatic macro-invertebrates from experimental freshwater ponds ("cattle tanks") with two different nutrient treatments.

**Details**

tank\_comm is a site-by-species matrix with individual counts.

tank\_plot\_attr is a data frame with corresponding site variables. The column group specifies whether a pond has received a "high" or "low" nutrient treatment. The columns x and y contain the spatial coordinates of the sites.

The data were adapted from Chase (2010).

**References**

Chase, J. M. (2010). Stochastic community assembly causes higher biodiversity in more productive environments. *Science*. 328:1388-1391.

**Examples**

```

data(tank_comm)
data(tank_plot_attr)
tank_mob_in = make_mob_in(tank_comm, tank_plot_attr)

```

# Index

- \* **burned**
  - fire\_comm, 20
- \* **cattle**
  - tank\_comm, 46
- \* **data**
  - fire\_comm, 20
  - inv\_comm, 30
  - tank\_comm, 46
- \* **fire**
  - fire\_comm, 20
- \* **invaded**
  - inv\_comm, 30
- \* **invasion**
  - inv\_comm, 30
- \* **ponds**
  - tank\_comm, 46
- \* **tanks**
  - tank\_comm, 46

avg\_nn\_dist, 2

calc\_beta\_div, 3

calc\_C\_target, 4, 8, 12, 13, 14, 17, 24

calc\_chao1, 5, 6, 9

calc\_comm\_div, 6, 7, 12, 25, 38

calc\_comm\_div\_ci, 10, 25, 28

calc\_div, 8, 12, 13, 24

calc\_PIE, 15, 17

calc\_S\_C, 17, 29

calc\_SPIE, 16, 16

Chat, 18

compare\_samp\_rarefaction, 19

fire\_comm, 20

fire\_plot\_attr (fire\_comm), 20

get\_delta\_stats, 20

get\_mob\_stats, 23, 35, 36

get\_null\_comm, 26

get\_samples, 12, 27

inv\_comm, 30

inv\_plot\_attr (inv\_comm), 30

invChat, 18, 29

kNCN\_average, 30

make\_mob\_in, 32, 45

makeCluster, 24, 31

mobr, 33

mobr-package (mobr), 33

plot.mob\_out, 34

plot.mob\_stats, 35

plot\_abu, 36

plot\_comm\_div, 37

plot\_N, 39

plot\_rarefaction, 39

rarefaction, 22, 41, 41

subset.mob\_in, 45

tank\_comm, 46

tank\_plot\_attr (tank\_comm), 46