Package ‘partitionBEFsp’

October 14, 2022

Title Methods for Calculating the Loreau & Hector 2001 BEF Partition

Version 1.0

Author Adam Clark

Maintainer Adam Clark <`adam.tclark@gmail.com`>

Description
A collection of functions that can be used to estimate selection and complementarity effects, sensu Loreau & Hector (2001) <doi:10.1038/35083573>, even in cases where data are only available for a random subset of species (i.e. incomplete sample-level data). A full derivation and explanation of the statistical corrections used here is available in Clark et al. (2019) <doi:10.1111/2041-210X.13285>.

Depends R (>= 3.4)

Imports graphics, stats

License GPL-3

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-21 08:20:05 UTC

R topics documented:

`calculate_DRY` *Calculate change in relative yield*

Description

calculates change in relative yield, DRY, comparing observed relative yield to the expected yield 1/Q

Usage

```
calculate_DRY(P, M, Q = length(M))
```

Arguments

`P`	biomass of species grown in polyculture
`M`	biomass of species grow in monoculture - note, must include the same species as P, listed in the same order
`Q`	number of species in the community -defaults to length(M), but note that if you are calculating DRY for a large community of Q species of which only N are observed, you should set Q=Q, rather than Q=N.

Value

a list of changes in relative yields

Examples

```
# Please see package help file (?partitionBEFsp) for examples.
```

`classic_partition` *Calculate classic partition*

Description

calculates the classic selection and complementarity effects, sensu Loreau and Hector 2001

Usage

```
classic_partition(DRY, M, N = length(M), Q = N,
  uncorrected_cov = FALSE)
```

Arguments

`DRY`	change in relative yield, as calculated by the calculate_DRY function
`M`	monoculture biomass
`N`	number of species in the sample - defaults to length(M)
`Q`	number of species in the full population - defaults to N - only required if uncorrected_cov="COMP"
`uncorrected_cov`	A character, which can be TRUE, FALSE, or COMP. Tells whether to use the standard sample-size corrected covariance function (FALSE), or a covariance function that is not corrected for sample size (TRUE), or a "compromise" function that resembles the standard function for N < Q, and that resembles the non-corrected function for N ~ Q If TRUE, then SS + CS = YO - YE, sensu Loreau and Hector 2001 defaults to FALSE note - we do not recommend setting this to TRUE or "COMP", unless you require SS+CS=YO-YE

Value

a list with elements S (the selection effect) and C (the complementarity effect)

`partitionBEFsp` *partitionBEFsp: A package for calculating the Loreau & Hector 2001 BEF partition.*

Description

The partitionBEFsp (or "partitioning Biodiversity-Ecosystem Functioning as sample-level and population-level estimates" package) is a collection of functions that can be used to estimate selection and complementarity effects, sensu Loreau & Hector 2001 (Nature 412:72-76), even in cases where data are only available for a random subset of species (i.e. incomplete sample-level data). A full derivation and explanation of the statistical corrections used here is available in Clark et al. 2019, Estimating Complementarity and Selection Effects from an Incomplete Sample of Species.

Source

Loreau, M., and Hector, A. (2001). Partitioning selection and complementarity in biodiversity experiments. Nature 412:72-76.

Examples

```
#Monoculture biomasses for 57 species
M<-c(57.57, 2.33, 306.25, 172.42, 351.48, 280.15, 216.93,
     1.30, 397.73, 185.57, 19.81, 162.45, 36.23, 42.48,
     3.16, 250.12, 5.30, 58.06, 172.93, 210.50, 253.78,
     15.96, 218.62, 282.00, 342.73, 242.18, 49.39, 100.00,
     112.20, 181.50, 61.98, 428.82, 911.55, 80.60, 206.75,
     108.25, 58.45, 154.55, 114.58, 144.38, 273.98, 25.41,
```

```
     148.82, 48.27, 35.62, 168.45, 157.98, 100.47, 31.12,
     9.86, 247.57, 182.32, 16.20, 251.30, 118.73, 137.65,
     149.93)

#Polyculture biomasses for a community of 57 species
P<-c(31.82, 0.06, 6.93, 6.75, 0.00, 0.11, 0.00,
     10.95, 0.19, 0.58, 0.01, 0.52, 21.72, 16.20,
     0.00, 0.09, 3.42, 0.00, 0.02, 3.18, 8.86,
     0.03, 0.02, 0.00, 10.14, 8.93, 4.53, 0.00,
     0.00, 0.02, 8.80, 0.31, 21.47, 0.34, 14.52,
     0.15, 0.00, 17.17, 66.55, 1.65, 0.44, 0.17,
     7.11, 0.45, 5.37, 7.66, 4.37, 0.00, 120.08,
     144.61, 0.00, 0.00, 0.00, 8.33, 93.18, 0.58,
     1.77)


#calculate DRY
DRY<-calculate_DRY(M=M, P=P, Q=length(M))

####################################
# Example 1: Classic partition
####################################

#calculate classic partition for full community
classic_partition(DRY=DRY, M=M)

#note that sum of partition equals the change in yield,
#but only if sample-size corrected covariance isn't used
N<-length(M)
cp_F<-classic_partition(DRY=DRY, M=M, uncorrected_cov = FALSE)
cp_T<-classic_partition(DRY=DRY, M=M, uncorrected_cov = TRUE)
cp_C<-classic_partition(DRY=DRY, M=M, uncorrected_cov = "COMP")

sum(P-M/N) #observed - expected yield
cp_F$S+cp_F$C #default
cp_T$S+cp_T$C #non-sample-size corrected
cp_C$S+cp_C$C #compromise

#also note that compromise only perfectly equals change in yield
#if Q = N (i.e. if the entire community is sampled)

sum(unlist(classic_partition(DRY=DRY, M=M, uncorrected_cov = "COMP", N=length(DRY), Q=N)))
sum(unlist(classic_partition(DRY=DRY, M=M, uncorrected_cov = "COMP", N=length(DRY), Q=N*2)))


####################################
# Example 2: Estimate population-level statistics
####################################
#estimate population-level partition for full community using only 30 species
set.seed(25123)
smp<-sample(30)
DRY_sample<-DRY[smp]
M_sample<-M[smp]
```

```
sample_to_population_partition(DRY=DRY_sample, M=M_sample, N=length(M_sample), Q=57)
#note - SP and CP are relatively close to the classic partition for the full community,
#whereas SS and CS are not.

#Repeat procedure for samples of between 2 and 57 species:
N_sample<-2:57
SP_est<-numeric(length(N_sample))
CP_est<-numeric(length(N_sample))

for(i in 1:length(N_sample)) {
  #sample N random species
  smp<-sample(1:57, N_sample[i])

  pop_est<-sample_to_population_partition(DRY=DRY[smp], M=M[smp], N=N_sample[i], Q=57)
  SP_est[i]<-pop_est$SP
  CP_est[i]<-pop_est$CP
}

#Plot estimates vs. true value (dotted line)
plot(N_sample, SP_est, type="b"); abline(h=classic_partition(DRY=DRY, M=M)$S, lty=3, col=2)
plot(N_sample, CP_est, type="b"); abline(h=classic_partition(DRY=DRY, M=M)$C, lty=3, col=2)
#note - estimates are noisy, but converge to the true value as N approaches Q.


####################################
# Example 3: Estimate sample-level statistics
####################################

#estimate expected value of sample-level statistics for a random sample of 30 species
#based on the full population of Q species
population_to_sample_partition(DRY=DRY, M=M, N=30, Q=57)

#Repeat procedure for samples of between 2 and 57 species:
N_sample<-2:57
SS_est<-numeric(length(N_sample))
CS_est<-numeric(length(N_sample))

for(i in 1:length(N_sample)) {
  pop_est<-population_to_sample_partition(DRY=DRY, M=M, N=N_sample[i], Q=57)
  SS_est[i]<-pop_est$SS
  CS_est[i]<-pop_est$CS
}

#Plot estimates vs. true value (dotted line)
plot(N_sample, SS_est/N_sample, type="b")
abline(h=classic_partition(DRY=DRY, M=M)$S/57, lty=3, col=2)
#note - expected value of SS/N = SP/Q for all N
plot(N_sample, CS_est/N_sample, type="b")
abline(h=classic_partition(DRY=DRY, M=M)$C/57, lty=3, col=2)
#note - expected value of CS/N is a biased estimate of SP/Q, especially for small N


####################################
```

```
# Example 4: Estimate confidence intervals
####################################
smp_ci<-sample_to_population_partition(DRY=DRY, M=M, Q=57, nboot=1000)
smp_ci$confint$bootdat_summary
```

---

population_to_sample_partition

Calculate sample-level partition

Description

takes a complete sample of all Q species in a community, and estimates sample-level selection and complementarity effects expected from a subset of N species drawn randomly from that community

Usage

```
population_to_sample_partition(DRY, M, N, Q = length(M),
  smallQ_correction = TRUE, uncorrected_cov = FALSE)
```

Arguments

`DRY`	change in relative yield, as calculated by the calculate_DRY function
`M`	monoculture biomass
`N`	number of species in the sample of the full community (i.e. the "sample") - defaults to length(M)
`Q`	total number of species in the full community (i.e. the "population")
`smallQ_correction`	tells whether to apply the correction for small Q, as shown in Eq. 3c in the main text - defaults to TRUE
`uncorrected_cov`	A character, which can be TRUE, FALSE, or COMP. Tells whether to use the standard sample-size corrected covariance function (FALSE), or a covariance function that is not corrected for sample size (TRUE), or a "compromise" function that resembles the standard function for N < Q, and that resembles the non-corrected function for N ~ Q If TRUE, then SS + CS = YO - YE, sensu Loreau and Hector 2001 defaults to FALSE note - we do not recommend setting this to TRUE or "COMP", unless you require SS+CS=YO-YE

Value

a list with elements SS (the sample-level selection effect), CS (the sample-level complementarity effect), SP (the population-level selection effect), and CP (the population-level complementarity effect),

Examples

```
# Please see package help file (?partitionBEFsp) for examples.
```

`sample_to_population_partition` *Calculate population-level partition*

Description

takes a random but incomplete sample of species of size N from a larger community Q, and estiamtes population-level selection and complementarity effects

Usage

```
sample_to_population_partition(DRY, M, N = length(M), Q,
  smallQ_correction = TRUE, uncorrected_cov = FALSE, nboot = NA)
```

Arguments

`DRY`	change in relative yield, as calculated by the calculate_DRY function
`M`	monoculture biomass
`N`	number of species in the sample of the full community (i.e. the "sample") - defaults to length(M)
`Q`	total number of species in the full community (i.e. the "population")
`smallQ_correction`	tells whether to apply the correction for small Q, as shown in Eq. 3c in the main text - defaults to TRUE
`uncorrected_cov`	A character, which can be TRUE, FALSE, or COMP. Tells whether to use the standard sample-size corrected covariance function (FALSE), or
`nboot`	Number of bootstrap iterations to run for estimating confidence intervals for selection and complementarity effects. Defaults to NA - i.e. no bootstrapping. a covariance function that is not corrected for sample size (TRUE), or a "compromise" function that resembles the standard function for N < Q, and that resembles the non-corrected function for N ~ Q If TRUE, then SS + CS = YO - YE, sensu Loreau and Hector 2001 defaults to FALSE note - we do not recommend setting this to TRUE or "COMP", unless you require SS+CS=YO-YE

Value

a list with elements SS (the sample-level selection effect), CS (the sample-level complementarity effect), SP (the population-level selection effect), CP (the population-level complementarity effect), and confint, which is a list that includes summary data and the full bootstrapped for estimates of the confidence intervals (if nboot != NA)

Examples

```
# Please see package help file (?partitionBEFsp) for examples.
```

Index