

SpatCourse_diversity

Stefano Larsen

3/17/2021

load packages

```
#install.packages('hillR')
library(hillR)

library(vegan)
library(tidyverse)
library(FD)

#install.packages('MeanRarity')
library(MeanRarity)
```

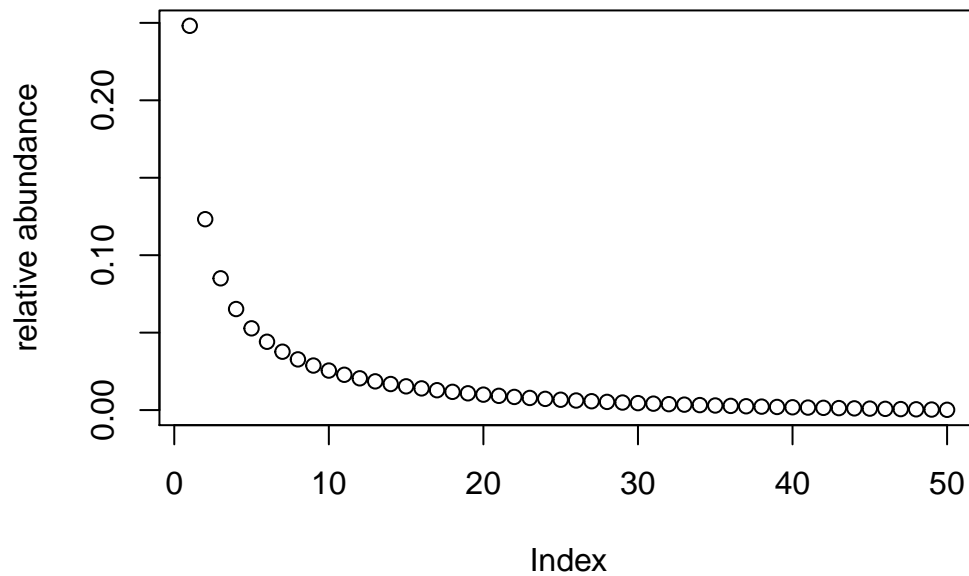
Diversity measures

We explore how the *classic* diversity measures relate to the *modern* hill-numbers based metrics, which reflect the *effective number of species* in the community

We first simulate a community with known SAD (species abundance distribution), and calculate taxon richness, and classic Shannon, and Simpson diversity. Then we remove half of the taxa (randomly) and re-quantify diversity metrics, using both classic and hill-numbers based diversity metrics.

```
# using the 'MeanRarity::fit_SAD' function we simulate community with 50 spp
#and known abundance distribution
test.community1=data.frame(t(fit_SAD(rich=50, simpson = 10)$rel_abundances))
colnames(test.community1)=paste("sp",1:50)
plot(t(test.community1), ylab='relative abundance', main='test community1')
```

test community1



let's calculate classic Shannon and Simpson diversity

```
diversity(test.community1, 'shannon')
```

```
## [1] 2.908134
```

```
diversity(test.community1, 'simpson')
```

```
## [1] 0.8999999
```

Now, lets remove 50% of species, randomly [like a large disturbance] Then calculate classic diversity for this halved community of 25 species

```
set.seed(1)
test.community2=sample(test.community1, 25); test.community2=sort(test.community2, decreasing = T)
diversity(test.community2, 'shannon')
```

```
## [1] 2.220554
```

```
diversity(test.community2, 'simpson')
```

```
## [1] 0.8123673
```

While the richness has declined by 50%, Shannon and Simpson diversity showed only modest decline;

Let's try the same using the *effective number of species* hill-based diversity

Hill-Shannon

```
renyi(test.community1, scale=1, hill=T) # this would be Shannon (q=1)
```

```
##          1
## 18.32258
## attr(,"class")
## [1] "renyi"    "numeric"
```

```
renyi(test.community2, scale=1, hill=T)
```

```
##          1
## 9.212432
## attr(,"class")
## [1] "renyi"    "numeric"
```

So hill-Shannon declines by ~50% (from c.18 to c.9), mirroring the decline in species number;
Let's see Hill-Simpson.

```
renyi(test.community1, scale=2, hill=T) # this would be Simpson (q=2)
```

```
##          2
## 9.99999
## attr(,"class")
## [1] "renyi"    "numeric"
```

```
renyi(test.community2, scale=2, hill=T)
```

```
##          2
## 5.329563
## attr(,"class")
## [1] "renyi"    "numeric"
```

Hill-Simpson also declined by ~50% (from c.10 to c.5).

The Hill-based conversion linearises the relation between taxonomic richness and diversity, making them more comparable.

We can visualise this simulating a community with 100 species **with equal abundance = 1 ind.** Call it 'comm1'

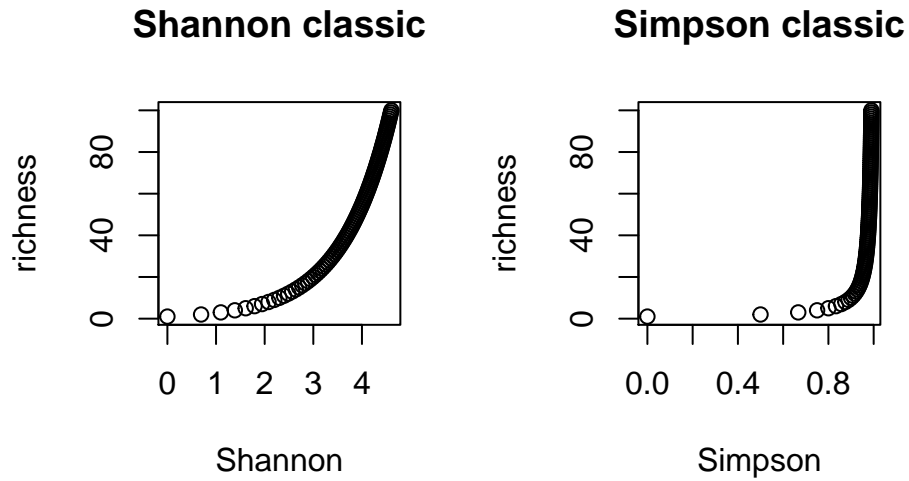
```
comm1=data.frame(t(rep(1,100))); colnames(comm1)=paste("sp",1:100)# simulated community of 100 spp

# calculate shannon classic and hill-shannon for each species in the community
shannon.test=data.frame(matrix(ncol=5, nrow=100))# matrix to hold the values
shannon.test[,1]=seq(1,100,1)
for(i in 1:100) shannon.test[i,2]= diversity(comm1[1:i], 'shannon')
for(i in 1:100) shannon.test[i,3]= diversity(comm1[1:i], 'simpson')
for(i in 1:100) shannon.test[i,4]= renyi(comm1[1:i], scale=1, hill=T)
for(i in 1:100) shannon.test[i,5]= renyi(comm1[1:i], scale=2, hill=T)

names(shannon.test)=c ('richness', 'Shannon', 'Simpson', 'Hill_Shannon', 'Hill_Simpson')
```

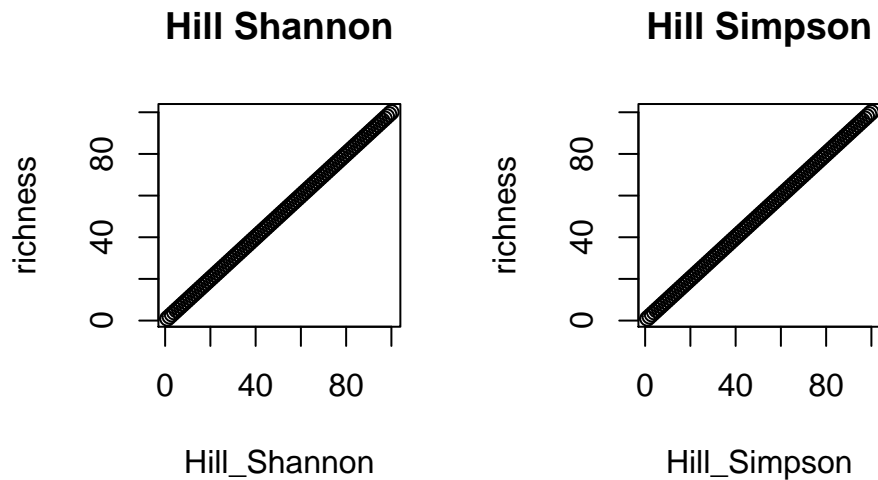
Plot classic Shannon & Simpson vs Richness.

```
par(mfrow=c(1,2))
plot(richness~Shannon, shannon.test, main='Shannon classic')
plot(richness~Simpson, shannon.test, main='Simpson classic')
```



Plot Hill-Shannon & Hill-Simpson vs Richness.

```
par(mfrow=c(1,2))
plot(richness~Hill_Shannon, shannon.test, main='Hill Shannon')
plot(richness~Hill_Simpson, shannon.test, main='Hill Simpson')
```



Exercises

The students are invited to calculate Shannon and Simpson index on community matrices already available in R using both the *classic* and *Hill* approach. You could use data such as `data(dune)` or `data(BCI)`. These are datasets on meadow vegetation species cover (dune) or tree counts in the Barro Colorado Island (BCI). **In both datasets, rows=sites and cols=species.**

TIP

to load a data object associated with e.g. the vegan package:

```
data(dune)
head(dune)
```

```
## Achimill Agrostol Airaprae Alop geni Anthodor Bellpere Bromhord Chenalbu
## 1      1      0      0      0      0      0      0      0
## 2      3      0      0      2      0      3      4      0
## 3      0      4      0      7      0      2      0      0
## 4      0      8      0      2      0      2      3      0
## 5      2      0      0      0      4      2      2      0
## 6      2      0      0      0      3      0      0      0
## Cirsarve Comapalu Eleopal u Elymrepe Empenigr Hyporadi Juncarti Juncbufo
## 1      0      0      0      4      0      0      0      0
## 2      0      0      0      4      0      0      0      0
## 3      0      0      0      4      0      0      0      0
## 4      2      0      0      4      0      0      0      0
## 5      0      0      0      4      0      0      0      0
## 6      0      0      0      0      0      0      0      0
## Lolipere Planlanc Poaprat Poatriv Ranuflam Rumeacet Sagipro c Salirepe
## 1      7      0      4      2      0      0      0      0
## 2      5      0      4      7      0      0      0      0
## 3      6      0      5      6      0      0      0      0
## 4      5      0      4      5      0      0      5      0
## 5      2      5      2      6      0      5      0      0
## 6      6      5      3      4      0      6      0      0
## Scora utu Trifprat Trifrepe Vicilath Bracruta Callcusp
## 1      0      0      0      0      0      0
## 2      5      0      5      0      0      0
## 3      2      0      2      0      2      0
## 4      2      0      1      0      2      0
## 5      3      2      2      0      2      0
## 6      3      5      5      0      6      0
```

Then, create a dataframe that holds information on the number of species, and the diversity indices for each site. You can then *remove* 50% of the species and calculate again the diversity indices. Make note of the differences. What are the proportional changes in classic and Hill-based indices?

TIP

to create a dataframe with e.g. number of species and shannon as columns using the dune data

```
my.df=data.frame(richness=specnumber(dune),
                 shannon=diversity(dune, 'shannon')
                 )
head(my.df)
```

```
##   richness  shannon
## 1         5 1.440482
## 2        10 2.252516
## 3        10 2.193749
## 4        13 2.426779
## 5        14 2.544421
## 6        11 2.345946
```

TIP

to randomly remove 50% of the species from the dune data, use *sample*

```
half_dune=dune[,c(sample(1:30, 15))]
```

You can compare the mean values of each indices in the original and *halved* dataset using `colMeans()`

The same approach can be used for quantifying functional diversity

This can be done using the ‘*hillR*’ package

First, verify that diversity from the ‘*hillR*’ package are the same as those calculated with `vegan::renyi`

```
renyi(test.community1, scale=1, hill=T)
```

```
##           1
## 18.32258
## attr(,"class")
## [1] "renyi"    "numeric"
```

```
hill_taxa(test.community1, q=1)
```

```
## [1] 18.32258
```

Yes! GOOD; Now lets see how it works on dummy communities with functional traits as well.

We can generate some fake trait values for the `comm1`.

```
set.seed(1)
comm1.trt=data.frame(trt1=rpois(100, 500), trt2=sample(1:100, 100), trt3=rgamma(100, 2,1))
row.names(comm1.trt)= colnames(comm1)# need to provide the same sp names on the trait data
```

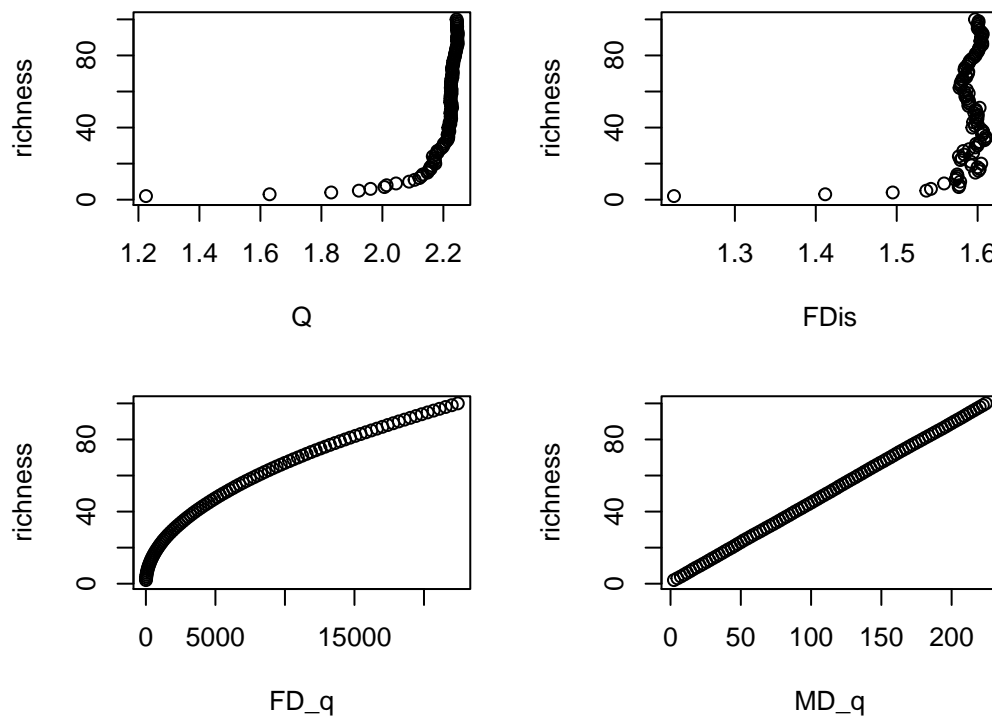
Calculate the range of functional diversities provided by *hillR*, over each richness value

```
# calculate some different functional diversity metrics; use q=1 to express it as hill-shannon
FD.test=data.frame(matrix(ncol=6, nrow=100))# matrix to hold the values
FD.test[,1]=seq(1,100,1)
for(i in 2:100) FD.test[i,2]= hill_func(comm1[1:i], comm1.trt[c(1:i),], q=1)[1]# Q
for(i in 2:100) FD.test[i,3]= hill_func(comm1[1:i], comm1.trt[c(1:i),], q=1)[2]# FDis
for(i in 2:100) FD.test[i,4]= hill_func(comm1[1:i], comm1.trt[c(1:i),], q=1)[3]# D_q
for(i in 2:100) FD.test[i,5]= hill_func(comm1[1:i], comm1.trt[c(1:i),], q=1)[4]# MD_q
for(i in 2:100) FD.test[i,6]= hill_func(comm1[1:i], comm1.trt[c(1:i),], q=1)[5]# FD_q
names(FD.test)=c ('richness', 'Q', 'FDis', 'D_q', 'MD_q', 'FD_q')
```

Let's plot the relation between number of taxa and functional diversities

```
par(mfrow=c(2,2), mar = c(4, 4, 2, 2))

plot(richness~Q, data=FD.test)
plot(richness~FDis, data=FD.test)
plot(richness~FD_q, FD.test)
plot(richness~MD_q, FD.test)
```



Here the results of course depends on the trait values of the species.

However, different patterns emerge looking at classic functional diversity measure (Q, FDis) relative to hill-based diversity measures (D_q: functional hill number, the effective number of equally abundant and functionally equally distinct species; FD_q: total functional diversity, the effective total functional distance between species of the assemblage). Refer to *Chiu & Chao (2014) PLoS ONE* for details on these metrics.