DiversitySeq: measuring diversity from count data sets

Francesca Finotello, Eleonora Mastrorilli, Barbara Di Camillo

September 2, 2018

Abstract

Next-generation sequencing, and particularly 16S ribosomal RNA (16S rRNA) gene sequencing, is a powerful technique for the identification and quantification of human-resident microbes, collectively known as the *human microbiota*. Once bacterial abundances are profiled via 16S rRNA gene sequencing and summarized in a count data set, diversity indices provide valuable mathematical tools to investigate the composition of the human microbiota. In brief, alpha diversity can be used to describe the taxonomical complexity of a single sample, whereas beta diversity can be used to identify differences between samples. The DiversitySeq package implements in a unified framework the whole panel of diversity indices reviewed in [1], enabling the assessment of diversity from count data sets. DiversitySeq also implements a simulator for the generation of synthetic count data sets from 16S rRNA gene sequencing. Besides 16S rRNA gene sequencing data, this package can be employed with other data sets with similar characteristics, such as 5S rRNA gene sequencing, environmental metagenomics or, more generally, any kind of matrix were counts are computed for different non-overlapping classes.

DiversitySeq **version:** 1.0

If you use DiversitySeq in published research, please cite:

F. Finotello, E. Mastrorilli, B. Di Camillo: **Measuring the diversity of the human microbiota with targeted next-generation sequencing**. Briefings in Bioinformatics 19 (4), 679-692, 2018.

Contents

1 Install and load the DiversitySeq package

This software is written in R language, so R must be installed on your computer to run DiversitySeq. For more information about the R environment, please refer to http://www.r-project.org/. DiversitySeq also depends on the package vegan, which can be installed with the following instruction:

```
install.packages("vegan", repos = "http://cran.us.r-project.org")
```

Once R and vegan are installed on your computer, DiversitySeq can be downloaded and installed as follows:

```
install.packages("http://sysbiobig.dei.unipd.it/?q=system/files/software/DiversitySeq_1.0.tar.gz",
repos = NULL, type = "source")
```

Alternatively, DiversitySeq can be also installed from a local archive file (.tar.gz) with the following command:

```
install.packages("Path_to_DiversitySeq/DiversitySeq_1.0.tar.gz",
repos=NULL, type="source")
```

Once installed, DiversitySeq can be loaded in the R environment by typing:

```
library("DiversitySeq")
```

DiversitySeq citation can be seen by using the command:

```
citation("DiversitySeq")
```

2 Compute alpha and beta diversity

The simulated 16S rRNA gene sequencing data generated in [1] can be used as test data to apply alpha and beta diversity indices. This count data set consists in a matrix of counts simCounts over 8,048 Operational Taxonomic Units (OTUs) and 20 samples. Data features were extracted from a real data set of the Human Microbiome Project [2], generated from stool samples.

```
data(stoolSimData)

dim(simCounts)

## [1] 8048   20

head(simCounts)

##              Sample_1 Sample_2 Sample_3 Sample_4 Sample_5 Sample_6 Sample_7 Sample_8
## OTU_97.10           0        0        0        0        0        0        0        0
## OTU_97.10002        0        0        0        0        0        0        0        0
## OTU_97.10005        4        0        0        0        1        0        0        0
## OTU_97.10008        1        1        0        1        0        0        0        0
## OTU_97.10010        0        0        0        0        0        0        0        0
## OTU_97.10016        0        0        0        0        0        0        0        0
##              Sample_9 Sample_10 Sample_11 Sample_12 Sample_13 Sample_14 Sample_15
## OTU_97.10           0         0         0         0         0         0         0
## OTU_97.10002        0         0         0         0         0         0         0
```

```
## OTU_97.10005         0         3         0         0         0         1         0
## OTU_97.10008         0         0         0         1        22         0         0
## OTU_97.10010         0         0         0         0         0         0         0
## OTU_97.10016         0         0         0         0         0         0         0
##              Sample_16 Sample_17 Sample_18 Sample_19 Sample_20
## OTU_97.10            0         0         0         0         0
## OTU_97.10002         0         0         0         0         0
## OTU_97.10005         8         0         2         0         0
## OTU_97.10008         1         0         1         1         0
## OTU_97.10010         0         0         0         0         0
## OTU_97.10016         0         0         0         0         0
```

Alpha diversity can be estimated using the `aindex` function, which takes as input the matrix of counts and a string indicating the diversity index to be used. The input matrix must be a matrix of species counts (here generally intended as read counts or species abundances), with species on the rows and samples on the columns. Here and throughout the vignette the term "species" is used for its ease of interpretation, but all discussions and computations can be intended considering any taxonomical level, such as *genera*, *phyla* or OTUs. The function allows the selection of various indices of diversity, richness and evenness through the parameter `index`. See Section 4 for a list of the selectable indices and [1] for their description and characterization. In the following example, alpha diversity is computed on the simulated data set using Shannon index.

```
stool.counts <- simCounts

stool.adiv <- aindex(stool.counts,
                     index = "Shannon")
```

In addition, `aindex` can consider the optional argument `group`: a vector indicating to which group the samples belong. The length of the `group` vector must equal the number of columns of the count matrix. When `group` is not specified, as in the example above, all samples are assigned to the same group, referred to as *group1*.

In the example below, the samples are split into two groups *groupA* and *groupB*.

```
stool.group.AB <- c(rep("groupA", ncol(stool.counts)/2),
    rep("groupB", ncol(stool.counts)/2))

stool.group.AB

##  [1] "groupA" "groupA" "groupA" "groupA" "groupA" "groupA" "groupA" "groupA" "groupA"
## [10] "groupA" "groupB" "groupB" "groupB" "groupB" "groupB" "groupB" "groupB" "groupB"
## [19] "groupB" "groupB"

stool.adiv.AB <- aindex(stool.counts,
                     index = "Shannon",
                     group = stool.group.AB)

names(stool.adiv.AB)

## [1] "groupA" "groupB"
```

As another example, alpha diversity can be computed with Hill numbers of order 2, assigning all samples to the same group *Stool*. The `q` parameter is used to specify the order of the diversity.

```
stool.group <- rep("Stool", ncol(stool.counts))

stool.adiv <- aindex(stool.counts,
                     index = "Hill", q = 2,
                     group = stool.group)
```

Similarly, beta diversity can be computed with the `bdiv` function, which allows the selection of various indices of beta diversity. See Section 4 for a list of the selectable indices and [1] for their description and characterization.

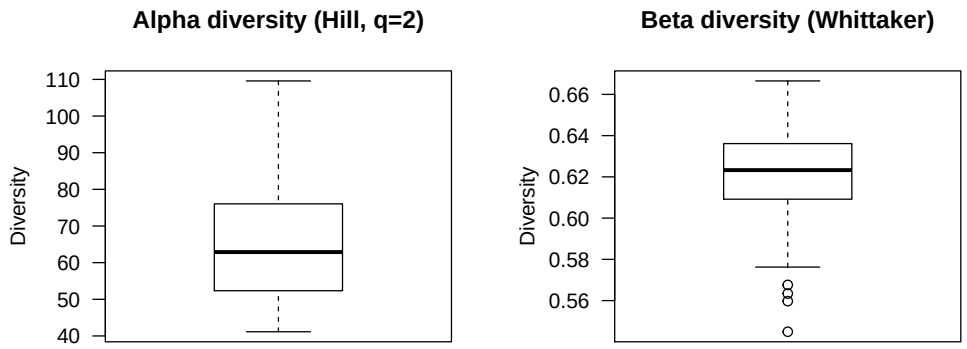
In the following example, beta diversity is computed on the simulated data set, employing Whittaker diversity and assigning all samples to the same group.

```
stool.bdiv <- bindex(stool.counts,
                     index = "w",
                     group = stool.group)
```

The output lists `stool.adiv` and `stool.bdiv` contain the values of alpha diversity for each sample and the values of beta diversity for each pair of samples, respectively, computed for each group (here only *Stool*).

Boxplots of alpha and beta diversity can be visualized employing the `divplot` function.

```
divplot(stool.adiv, main = "Alpha diversity (Hill, q=2)")
divplot(stool.bdiv, main = "Beta diversity (Whittaker)")
```



Diversity can be assessed in multiple groups or data sets using the functions described above. `DiversitySeq` contains an additional 16S rRNA gene sequencing data, the *Saliva* data set, simulated similarly to the *Stool* data set. The parameters for the simulation of the *Saliva* data set were extracted from a real data set of the Human Microbiome Project [2].

```
data(salivaSimData)

dim(simCounts)

## [1] 15094    20

saliva.counts <- simCounts
saliva.group <- rep("Saliva", ncol(saliva.counts))
```

The *Stool* and *Saliva* count matrices and group annotations can be now merged using the function `mergedatasets`, which takes as input two lists with the count data and groups to be merged. The count data sets and group vectors must appear in the two lists in the same order (here, *Stool* and then *Saliva*). The data sets to be merged can be more than two.

```
mrgData <- mergedatasets(list(stool.counts, saliva.counts),
                         list(stool.group, saliva.group))
```

The merged count matrix has a number of rows which is equal to the union of all species assayed in the merged data sets. Please notice that the `mergedatasets` function only works when row names in each count matrix are defined.

```
all.counts <- mrgData$data

dim(all.counts)

## [1] 21494    40

length(union(rownames(stool.counts),rownames(saliva.counts)))

## [1] 21494
```

The merged group vector comprises the 20 *Stool* samples and the 20 *Saliva* samples.

```
all.group <- mrgData$group

table(all.group)

## all.group
## Saliva  Stool
##     20     20
```

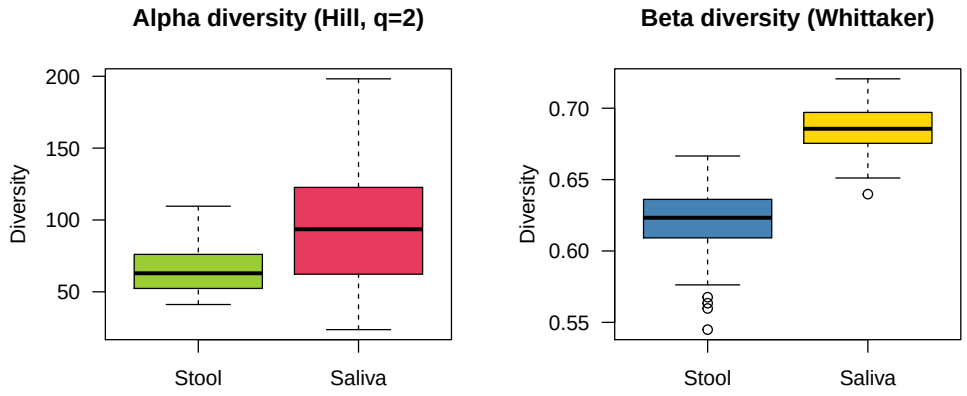
Alpha and beta diversity in the two groups can be now assessed and visualized as before. The parameter `col` of the `divplot` function allows specifying a default (for up to 4 groups) or a custom palette for the boxplots.

```
all.adiv <- aindex(all.counts,
                   index = "Hill", q = 2,
                   group = all.group)

all.bdiv <- bindex(all.counts,
                   index = "w",
                   group = all.group)

divplot(all.adiv, main = "Alpha diversity (Hill, q=2)",  col="default")

divplot(all.bdiv, main = "Beta diversity (Whittaker)", col=c("#4682B4","#FFD700"))
```



3 Simulate 16S rRNA gene sequencing data sets

It is possible to generate *ad hoc* count matrices using the `simcounts` function, which assumes a negative binomial (NB) model of data. In particular, the biological variability is modeled with a Gamma distribution with dispersion parameter `phi` and the technical variability is modeled with a Poisson distribution (for more details on the simulation model, please refer to [1]). The function takes as input a vector of average species abundances `avgAbund`, the coefficient of dispersion of the NB distribution `phi` and a vector of sequencing depths `sdepth` for the samples to be simulated.

As an example, we can use the parameters used for the simulated data in [1].

```
data(stoolSimData)
stool.avgAbundances<-avgAbundances
stool.phi<-phi
stool.sdepth<-sdepth

stool.simdataset <- simulatecounts(avgAbund = stool.avgAbundances,
                                   phi = stool.phi,
                                   sdepth = stool.sdepth)

## Simulating a count data set with 20 samples and 8048 species...
```

The output of the simulation is a list containing both the matrix of simulated counts and the matrix of the simulated abundances. From here on, we will refer to this new data set as *Stool2*. Please notice that the random nature of the simulation, coupled with the the limited sequencing depth employed, can lead to very different simulated data sets and, thus, estimated diversity. Therefore, the results and plots obtained with the following codes might differ significantly from the ones reported in the present vignette. Reproducibility improves with higher sequencing depths.

```
stool2.counts<-stool.simdataset$counts

stool2.abundances<-stool.simdataset$abundances

stool2.group<-rep("Stool2", ncol(stool2.counts))
```

The *Stool2* count data set can be merged with the previous ones. As an example, the indices of evenness (*EF*) and relative evenness (*RLE*) of order 2 can be now computed for the three groups. In the following

plots of diversity, the actual data points are also plotted over the boxplots specifying the `points` parameter in the `divplot` function.

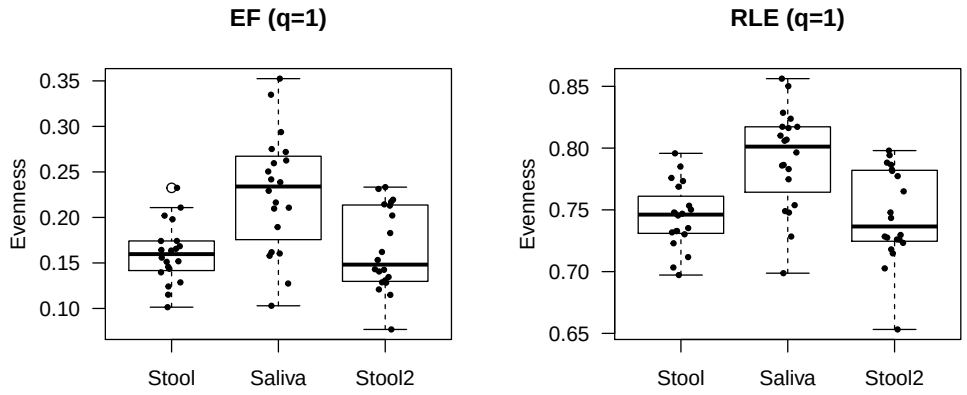
```
mrgData <- mergedatasets(list(all.counts, stool2.counts),
                         list(all.group, stool2.group))
all.counts <- mrgData$data
all.group <- mrgData$group

all.EF <- aindex(all.counts,
                     index = "EF", q = 1,
                     group = all.group)

all.RLE <- aindex(all.counts,
                     index = "RLE", q=1,
                     group = all.group)

divplot(all.EF, main = "EF (q=1)",  ylab = "Evenness",
    points = TRUE, cexpoints = 0.8)

divplot(all.RLE, main = "RLE (q=1)", ylab = "Evenness",
    points = TRUE, cexpoints = 0.8)
```



4 Available diversity indices

Tables 1 and 2 summarize all indices that can be employed with the `aindex` and `bindex` functions, together with: the string to be used as `index` parameter, a check mark indicating whether the `q` parameter is mandatory (only for alpha diversity) and the implemented mathematical formulation.

The symbols used in index formulae are reported in what follows (see [1] for a detailed description):

S^{obs}	observed richness (i.e. number of species with non-zero counts)
p_i	relative counts of the i-th species
q	order of the diversity
f_1	number of species with 1 count (*singletons*)
f_2	number of species with 2 counts (*doubletons*)
S^{rare}	number of *rare* species with no more than 10 counts
S^{abund}	number of *abundant* species with more than 10 counts
N^{rare}	total counts accounted by all rare species
C^{ACE}	proportion of counts accounted by singletons
γ^{ACE}	coefficient of variation of the rare species
a	number of species shared between two samples
b	number of species unique to the first sample
c	number of species unique to the second sample

Table 1. Alpha diversity indices that can be selected with the `aindex` function:

Index	Mandatory `q`	Formula
`Richness`		S^{obs}
`Chao1`		$S^{Chao1} = S^{obs} + \frac{f_1(f_1-1)}{2(f_2+1)}$
`Jackknife1`		$S^{Jackk1} = S^{obs} + f_1$
`Jackknife2`		$S^{Jackk2} = S^{obs} + 2f_1 - f_2$
`ACE`		$S^{ACE} = S^{abund} + \frac{S^{rare}}{C^{ACE}} + \frac{f_1}{C^{ACE}} + (\gamma^{ACE})^2$
`Hill`	✓	${}^qD = (\sum_{i=1}^{S^{obs}} p_i^q)^{\frac{1}{1-q}}$ for $q = 1$, ${}^1D = exp(-\sum_{i=1}^{S^{obs}} p_i \cdot ln(p_i))$
`BergerParker`		${}^\infty D = \frac{1}{max(p_i)}$
`Renyi`	✓	${}^qRE = \frac{1}{1-q} \cdot ln(\sum_{i=1}^{S^{obs}} p_i^q)$
`iSimpson` (1)		$IS = (\sum_{i=1}^{S^{obs}} {p_i}^2)^{-1}$
`cSimpson` (2)		$GS = 1 - \sum_{i=1}^{S^{obs}} {p_i}^2$
`Shannon`		$H = -\sum_{i=1}^{S^{obs}} p_i \cdot ln(p_i)$
`Tail`		$T = \sqrt{\sum_{i=1}^{S^{obs}} (i-1)^2 \cdot p_i}$
`EF`	✓	${}^qEF = \frac{{}^qD}{{}^0D}$
`IF`	✓	${}^qIF = \frac{{}^0D}{{}^qD}$
`RLE`	✓	${}^qRLE = \frac{ln({}^qD)}{ln({}^0D)}$
`RLI`	✓	${}^qRLI = 1 - \frac{ln({}^qD)}{ln({}^0D)}$
`Pielou`		$P = \frac{ln({}^1D)}{ln({}^0D)}$

Notes: (1)Inverse Simpson; (2)Complementary Simpson or Gini-Simpson.

Table 2. Beta diversity indices that can be selected with the `bindex` function:

Index	Formula
`w`	$\beta_w = \frac{b+c}{2a+b+c}$
`c`	$\beta_c = \frac{b+c}{2}$
`cc`	$\beta_{cc} = \frac{b+c}{a+b+c}$
`co`	$\beta_{co} = 1 - \frac{a \cdot (2a+b+c)}{(a+b)(a+c)}$
`m`	$\beta_m = \frac{(2a+b+c)(b+c)}{(a+b+c)}$
`mn`	$\beta_{mn} = \frac{(2a+b+c)(b+c)}{(a+b+c)^2}$
`rs`	$\beta_{rs} = \frac{2 \cdot (bc+1)}{(a+b+c)^2 - (a+b+c)}$
`r`	$\beta_r = \frac{2bc}{(a+b+c)^2 - 2bc}$
`-3`	$\beta_{-3} = \frac{min(b,c)}{a+b+c}$
`-3n`	$\beta_{-3n} = \frac{2 \cdot min(b,c)}{a+b+c}$
`-2`	$\beta_{-2} = \frac{min(b,c)}{a+max(b,c)}$
`sim`	$\beta_{sim} = \frac{min(b,c)}{a+min(b,c)}$
`I`	$\beta_I = log(2a+b+c) - \frac{2a \cdot log(2) + (a+b) \cdot log(a+b) + (a+c) \cdot log(a+c)}{2a+b+c}$
`e`	$\beta_e = exp\Big(log(2a+b+c) - \frac{2a \cdot log(2) + (a+b) \cdot log(a+b) + (a+c) \cdot log(a+c)}{2a+b+c}\Big) - 1$
`z`	$\beta_z = 1 - \frac{log(2a+b+c) - log(a+b+c)}{log(2)}$

5 Session Info

- R version 3.4.2 (2017-09-28), `x86_64-apple-darwin15.6.0`
- Locale: `en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Running under: `macOS High Sierra 10.13.6`
- Matrix products: default
- BLAS: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib`
- LAPACK: `/Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: DiversitySeq 1.0, knitr 1.18, lattice 0.20-35, permute 0.9-4, vegan 2.5-2
- Loaded via a namespace (and not attached): cluster 2.0.6, compiler 3.4.2, evaluate 0.10.1, grid 3.4.2, highr 0.6, magrittr 1.5, MASS 7.3-48, Matrix 1.2-12, mgcv 1.8-22, nlme 3.1-131, parallel 3.4.2, stringi 1.1.6, stringr 1.2.0, tools 3.4.2

References

[1] Francesca Finotello, Eleonora Mastrorilli, and Barbara Di Camillo. "Measuring the diversity of the human microbiota with targeted Next-Generation Sequencing". In: *Briefings in bioinformatics* 19.4 (2018), pp. 679–692.

[2] Jane Peterson et al. "The NIH human microbiome project". In: *Genome research* 19.12 (2009), pp. 2317–2323.