Alakazam: Analysis of clonal diversity

Jason Anthony Vander Heiden

2016-01-28

Contents

The clonal diversity of the repertoire can be analyzed using the general form of the diversity index, as proposed by Hill in:

```
Hill, M. Diversity and evenness: a unifying notation and its consequences.
    Ecology 54, 427-432 (1973).
```

Coupled with resampling strategies to correct for variations in sequencing depth, as well as inferrence of complete clonal abundance distributions as described in:

```
Chao A, et al. Rarefaction and extrapolation with Hill numbers:
    A framework for sampling and estimation in species diversity studies.
    Ecol Monogr. 2014 84:45-67.
Chao A, et al. Unveiling the species-rank abundance distribution by
    generalizing the Good-Turing sample coverage theory.
    Ecology. 2015 96, 11891201.
```

This package provides methods for the inferrence of a complete clonal abundance distribution, using the `estimateAbundance` function, along with two approaches to assess diversity of these distributions:

1. Generation of a smooth diversity (D) curve over a range of diversity orders (q) using `rarefyDiversity`.
2. A significance test of the diversity (D) at a fixed diversity order (q) using `testDiversity`.

Load Change-O data

A small example Change-O tab-delimited database file is included in the `alakazam` package. Diversity calculation requires the `CLONE` field (column) to be present in the Change-O file, as well as an additional grouping column. In this example we will use the grouping columns `SAMPLE` and `ISOTYPE`.

```
library(alakazam)

# Load Change-O file
file <- system.file("extdata", "changeo_demo.gz", package="alakazam")
df <- readChangeoDb(file)
```

Generate a clonal abundance curve

A simple table of the observed clonal abundance counts and frequencies may be generated using the `countClones` function either without copy numbers, where the size of each clone is determined by the number of sequence members:

```
# Partitions the data based on the SAMPLE column
clones <- countClones(df, groups="SAMPLE")
head(clones, 5)
```

```
## Source: local data frame [5 x 4]
## Groups: SAMPLE [1]
##
##   SAMPLE CLONE SEQ_COUNT SEQ_FREQ
##    (chr) (chr)     (int)    (dbl)
## 1   RL01    53         4     0.04
## 2   RL01    91         4     0.04
## 3   RL01     2         2     0.02
## 4   RL01    21         2     0.02
## 5   RL01    33         2     0.02
```

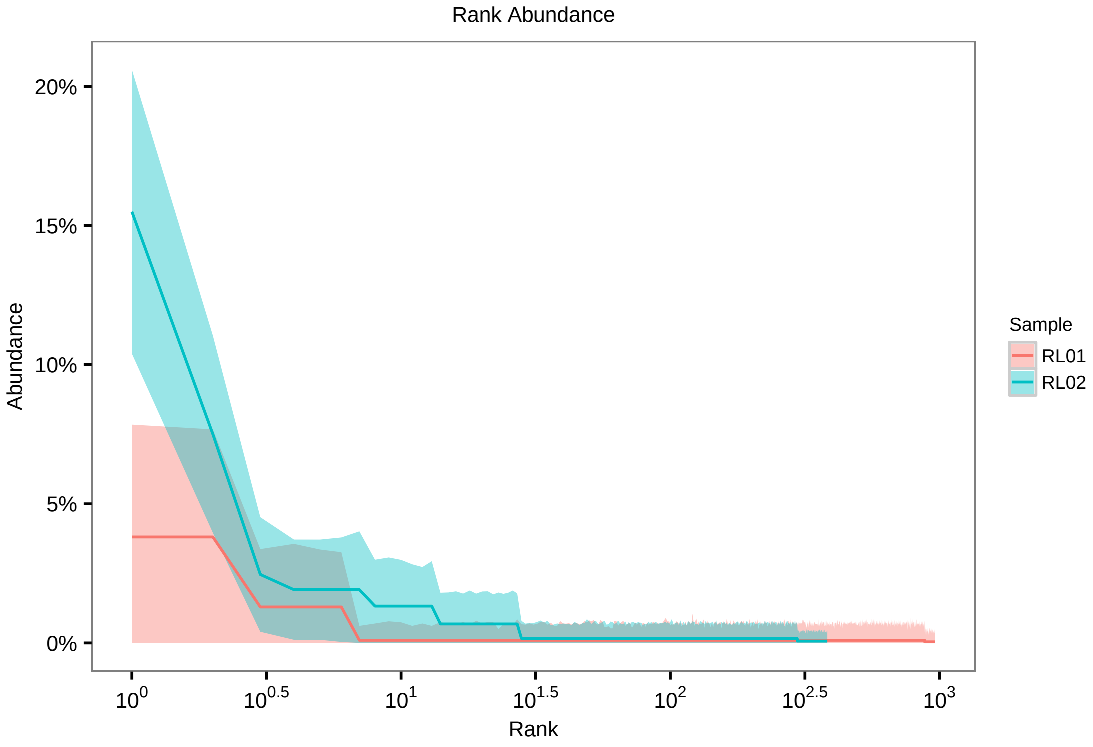
You may also specify a column containing the abundance count of each sequence (usually copy numbers), that will including weighting of each clone size by the corresponding abundance count. Furthermore, multiple grouping columns may be specified such that `SEQ_FREQ` (unwieghted clone size as a fraction of total sequences in the group) and `COPY_FREQ` (weighted faction) are normalized to within multiple group data partitions.

```
# Partitions the data based on both the SAMPLE and ISOTYPE columns
# Weights the clone sizes by the DUPCOUNT column
clones <- countClones(df, groups=c("SAMPLE", "ISOTYPE"), copy="DUPCOUNT")
head(clones, 5)
```

```
## Source: local data frame [5 x 7]
## Groups: SAMPLE, ISOTYPE [2]
##
##   SAMPLE ISOTYPE CLONE SEQ_COUNT COPY_COUNT SEQ_FREQ  COPY_FREQ
##    (chr)   (chr) (chr)     (int)      (int)    (dbl)      (dbl)
## 1   RL01     IgA    90         1         10      0.2 0.71428571
## 2   RL01     IgA     2         2          2      0.4 0.14285714
## 3   RL01     IgA     1         1          1      0.2 0.07142857
## 4   RL01     IgA    24         1          1      0.2 0.07142857
## 5   RL01     IgD    33         2          2      0.2 0.20000000
```

While `countClones` will report observed abundances, it will not correct the distribution nor provide confidence intervals. A complete clonal abundance distribution may be inferred using the `estimateAbundance` function with confidence intervals derived via bootstrapping. This output may be visualized using the `plotAbundance` function.

```
# Partitions the data on the SAMPLE column
# Calculates a 95% confidence interval via 200 bootstrap realizations
clones <- estimateAbundance(df, group="SAMPLE", ci=0.95, nboot=200)

head(clones, 5)

## Source: local data frame [5 x 6]
##
##    GROUP CLONE          P LOWER      UPPER  RANK
##    (chr) (chr)      (dbl) (dbl)      (dbl) (int)
## 1   RL01    53 0.03808654     0 0.07845412     1
## 2   RL01    91 0.03808654     0 0.07674582     2
## 3   RL01     2 0.01293068     0 0.03372504     3
## 4   RL01    21 0.01293068     0 0.03560444     4
## 5   RL01    33 0.01293068     0 0.03357504     5

# Plots a rank abundance curve of the relative clonal abundances
p1 <- plotAbundance(clones, legend_title="Sample")
```



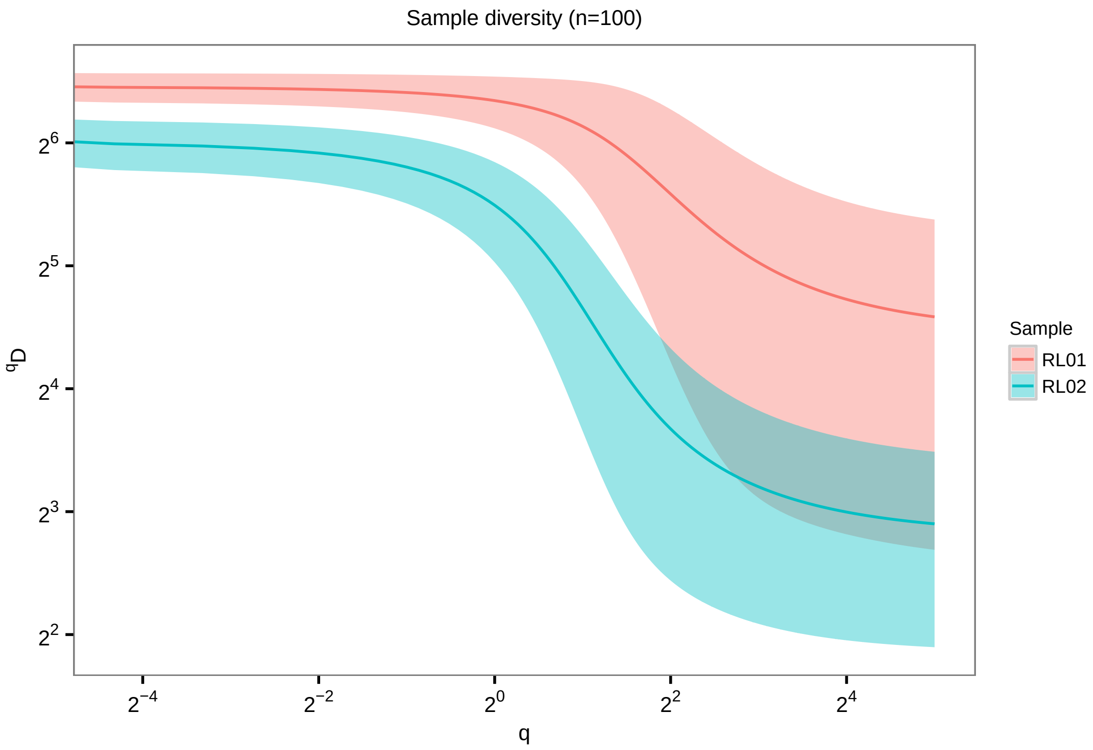
Generate a diversity curve

The function `rarefyDiversity` performs uniform resampling of the input sequences and recalculates the clone size distribution, and diversity, with each resampling realization. Diversity (D) is calculated over a range of diversity orders (q) to generate a smooth curve.

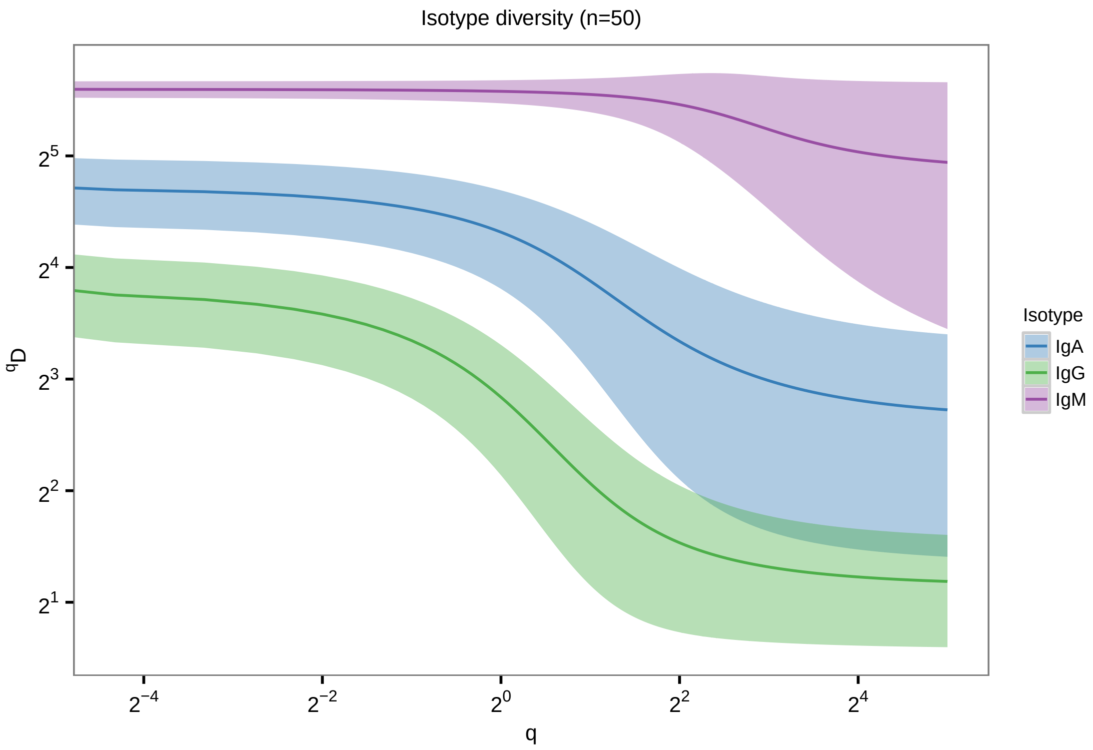
```r
# Compare diversity curve across values in the "SAMPLE" column
# q ranges from 0 (min_q=0) to 32 (max_q=32) in 0.05 incriments (step_q=0.05)
# A 95% confidence interval will be calculated (ci=0.95)
# 2000 resampling realizations are performed (nboot=200)
sample_div <- rarefyDiversity(df, "SAMPLE", min_q=0, max_q=32, step_q=0.05,
                                ci=0.95, nboot=200)

# Compare diversity curve across values in the "ISOTYPE" column
# Analyse is restricted to ISOTYPE values with at least 30 sequences by min_n=30
# Excluded groups are indicated by a warning message
isotype_div <- rarefyDiversity(df, "ISOTYPE", min_n=30, min_q=0, max_q=32,
                                 step_q=0.05, ci=0.95, nboot=200)
```

```
## Warning in rarefyDiversity(df, "ISOTYPE", min_n = 30, min_q = 0, max_q =
## 32, : Not all groups passed threshold min_n=30.Excluded: IgD
```

```r
# Plot a log-log (log_q=TRUE, log_d=TRUE) plot of sample diversity
# Indicate number of sequences resampled from each group in the title
sample_main <- paste0("Sample diversity (n=", sample_div@n, ")")
p2 <- plotDiversityCurve(sample_div, main_title=sample_main,
                         legend_title="Sample", log_q=TRUE, log_d=TRUE)
```



```
# Plot isotype diversity using default set of Ig isotype colors
isotype_main <- paste0("Isotype diversity (n=", isotype_div@n, ")")
p3 <- plotDiversityCurve(isotype_div, colors=IG_COLORS, main_title=isotype_main,
                         legend_title="Isotype", log_q=TRUE, log_d=TRUE)
```



Test diversity at a fixed diversity order

The function `testDiversity` performs resampling and diversity calculation in the same manner as `rarefyDiversity`, but only for a single diversity order. Significance testing across groups is performed using the delta of the bootstrap distributions between groups.

```
# Test diversity at q=0 (species richness) across values in the "SAMPLE" column
# 2000 bootstrap realizations are performed (nboot=200)
sample_test <- testDiversity(df, 0, "SAMPLE", nboot=200)

sample_test

## An object of class "DiversityTest"
## Slot "tests":
##           test pvalue delta_mean delta_sd
## 1 RL01 != RL02      0      24.99 5.354742
##
## Slot "summary":
##      group  mean       sd
## RL01  RL01 88.51 3.246939
## RL02  RL02 63.52 4.302576
##
## Slot "groups":
## [1] "RL01" "RL02"
```

```
##
## Slot "q":
## [1] 0
##
## Slot "n":
## RL01 RL02
##  100  100
##
## Slot "nboot":
## [1] 200
```

```
# Test diversity across values in the "ISOTYPE" column
# Analyse is restricted to ISOTYPE values with at least 30 sequences by min_n=30
# Excluded groups are indicated by a warning message
isotype_test <- testDiversity(df, 2, "ISOTYPE", min_n=30, nboot=200)
```

```
## Warning in testDiversity(df, 2, "ISOTYPE", min_n = 30, nboot = 200): Not
## all groups passed threshold min_n=30. Excluded: IgD
```

```
isotype_test
```

```
## An object of class "DiversityTest"
## Slot "tests":
##         test pvalue delta_mean delta_sd
## 1 IgA != IgG      0   10.20885 3.558916
## 2 IgA != IgM      0   32.47844 4.028593
## 3 IgG != IgM      0   42.68730 2.672437
##
## Slot "summary":
##     group      mean       sd
## IgA   IgA 14.556819 3.270013
## IgG   IgG  4.347964 1.056252
## IgM   IgM 47.035262 2.422166
##
## Slot "groups":
## [1] "IgA" "IgG" "IgM"
##
## Slot "q":
## [1] 2
##
## Slot "n":
## IgA IgG IgM
##  50  50  50
##
## Slot "nboot":
## [1] 200
```