Alakazam: Analysis of clonal abundance and diversity

Jason Anthony Vander Heiden

2019-07-17

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 `alphaDiversity`.
2. A significance test of the diversity (D) at a fixed diversity order (q).

Example data

A small example Change-O database, `ExampleDb`, 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`.

```r
# Load required packages
library(alakazam)

# Load example data
data(ExampleDb)
```

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(ExampleDb, group="SAMPLE")
head(clones, 5)
```

```
## # A tibble: 5 x 4
## # Groups:   SAMPLE [1]
##   SAMPLE CLONE SEQ_COUNT SEQ_FREQ
##   <chr>  <chr>     <int>    <dbl>
## 1 +7d    3128        100    0.1
## 2 +7d    3100         50    0.05
## 3 +7d    3141         44    0.044
## 4 +7d    3177         30    0.03
## 5 +7d    3170         28    0.028
```

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.

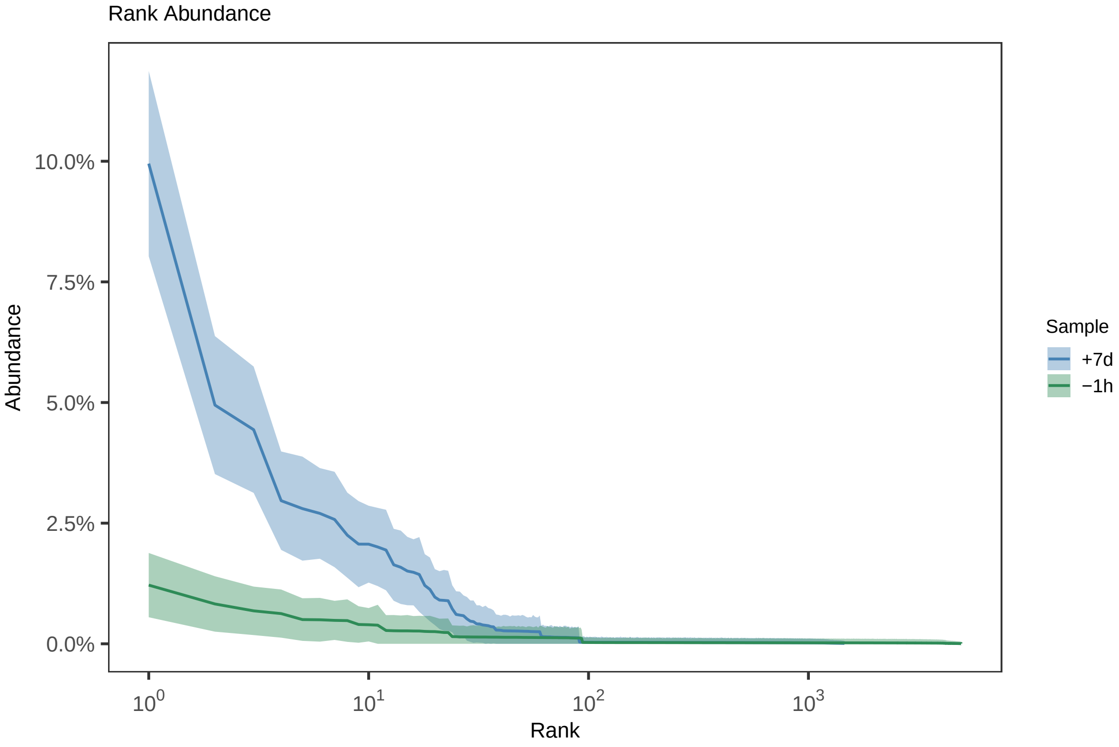
```
# Partitions the data based on both the SAMPLE and ISOTYPE columns
# Weights the clone sizes by the DUPCOUNT column
clones <- countClones(ExampleDb, group=c("SAMPLE", "ISOTYPE"), copy="DUPCOUNT")
head(clones, 5)
```

```
## # A tibble: 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 +7d    IgA     3128         88        651   0.331     0.497
## 2 +7d    IgG     3100         49        279   0.0928    0.173
## 3 +7d    IgA     3141         44        240   0.165     0.183
## 4 +7d    IgG     3192         19        141   0.0360    0.0874
## 5 +7d    IgG     3177         29        130   0.0549    0.0806
```

While `countClones` will report observed abundances, it will not 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 `plotAbundanceCurve` function.

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

# Plots a rank abundance curve of the relative clonal abundances
```

```
sample_colors <- c("-1h"="seagreen", "+7d"="steelblue")
plot(curve, colors = sample_colors, legend_title="Sample")
```

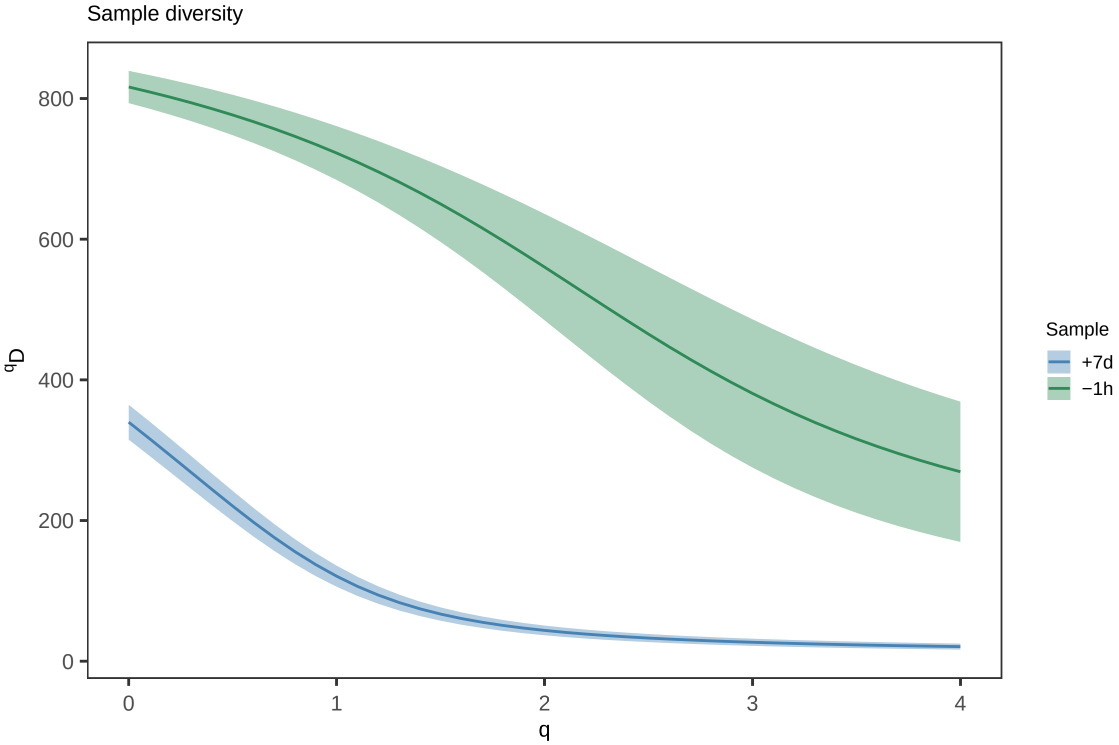


Generate a diversity curve

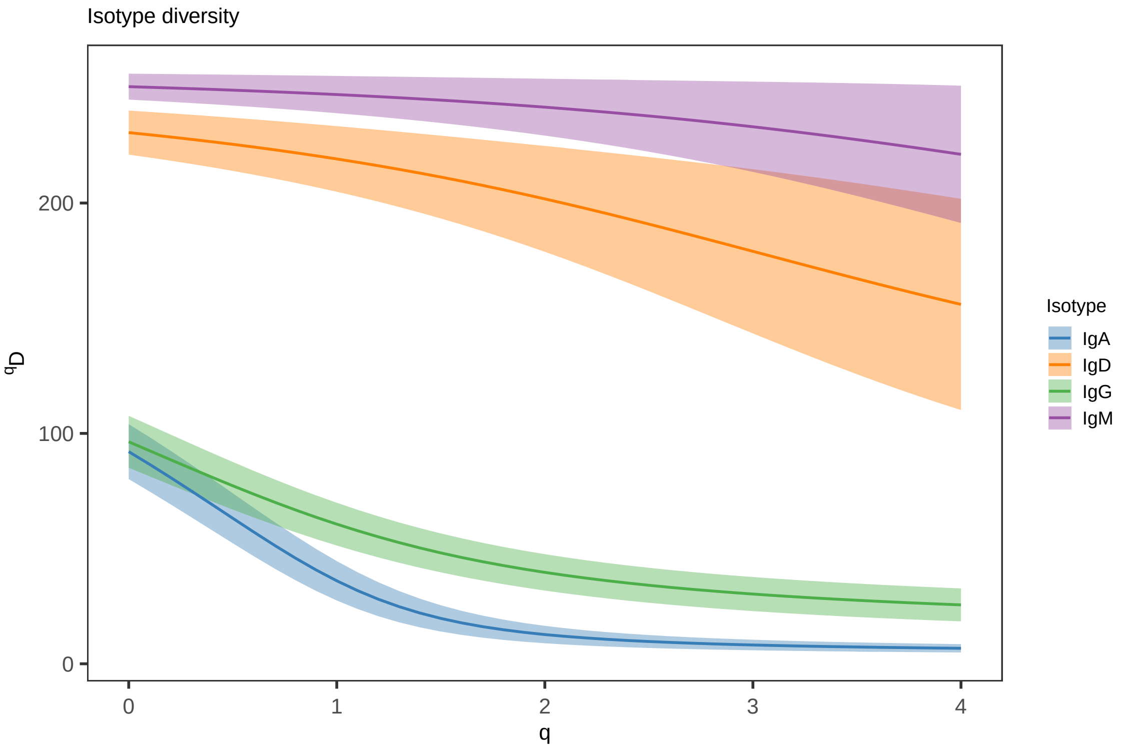
The function `alphaDiversity` performs uniform resampling of the input sequences and recalculates the clone size distribution, and diversity, with each resampling realization. Diversity (D) is calculated over a range of diversity orders (q) to generate a smooth curve.

```
# Compare diversity curve across values in the "SAMPLE" column
# q ranges from 0 (min_q=0) to 4 (max_q=4) in 0.05 incriments (step_q=0.05)
# A 95% confidence interval will be calculated (ci=0.95)
# 200 resampling realizations are performed (nboot=200)
sample_curve <- alphaDiversity(ExampleDb, group="SAMPLE",
                               min_q=0, max_q=4, step_q=0.1,
                               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_curve <- alphaDiversity(ExampleDb, group="ISOTYPE",
                                min_q=0, max_q=4, step_q=0.1,
                                ci=0.95, nboot=200)

# 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")
```

```
sample_colors <- c("-1h"="seagreen", "+7d"="steelblue")
plot(sample_curve, colors=sample_colors, main_title=sample_main,
     legend_title="Sample")
```



```
# Plot isotype diversity using default set of Ig isotype colors
isotype_main <- paste0("Isotype diversity")
plot(isotype_curve, colors=IG_COLORS, main_title=isotype_main,
     legend_title="Isotype")
```



View diversity tests at a fixed diversity order

Significance testing across groups is performed using the delta of the bootstrap distributions between groups when running `alphaDiversity` for all values of `q` specified.

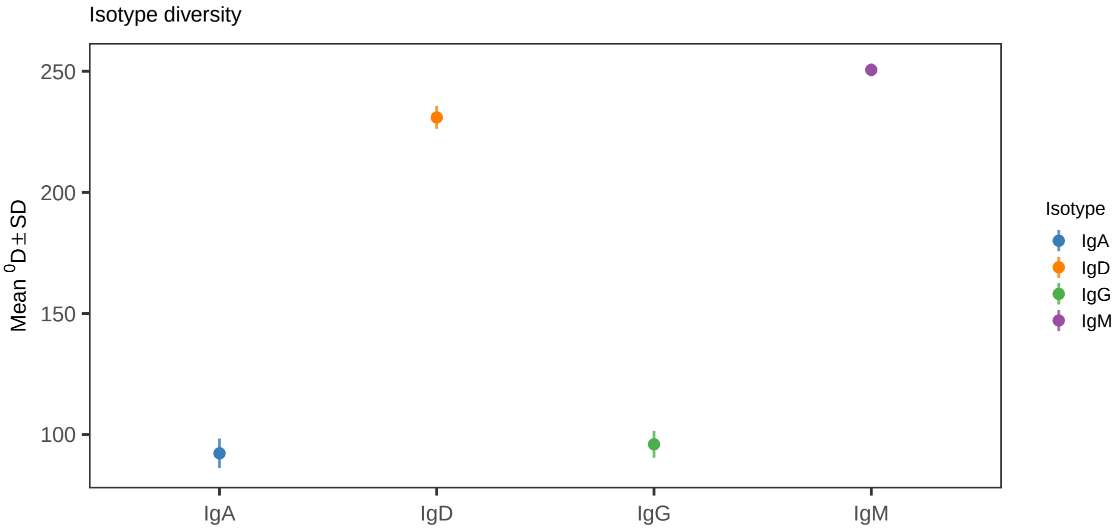
```
# Test diversity at q=0, q=1 and q=2 (equivalent to species richness, Shannon entropy,
# Simpson's index) across values in the "SAMPLE" column
# 200 bootstrap realizations are performed (nboot=200)
isotype_test <- alphaDiversity(ExampleDb, group="ISOTYPE", min_q=0, max_q=2, step_q=1, nboot=2
```

```
## [1] 0 1 2
```

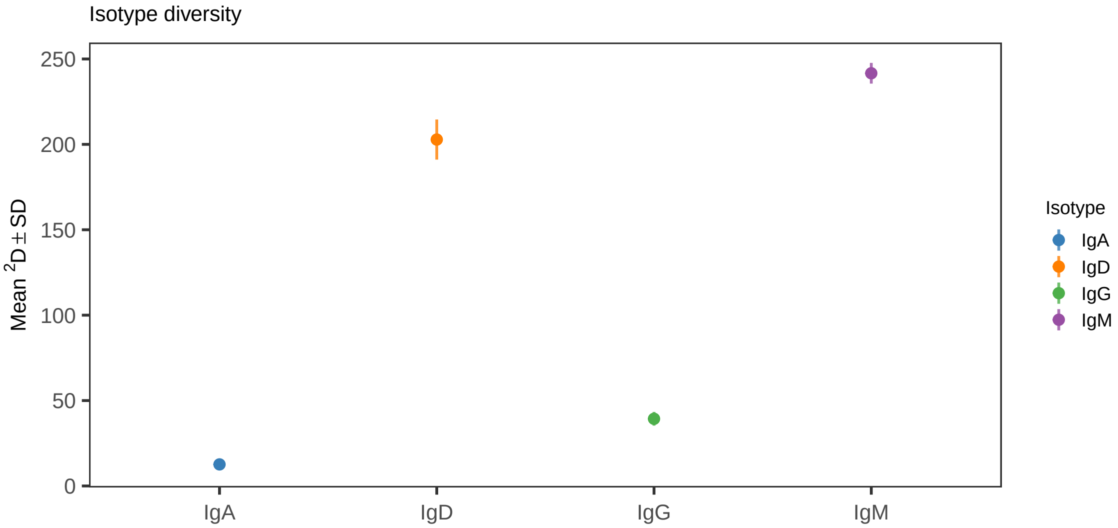
```
# Print P-value table
print(isotype_test)
```

```
## # A tibble: 12 x 9
## # Groups:   ISOTYPE [4]
##    ISOTYPE     Q     D  D_SD D_LOWER D_UPPER     E E_LOWER E_UPPER
##    <chr>   <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1 IgA         0  92.2  6.09   80.3    104.  1       0.871   1.13
##  2 IgA         1  36.0  3.83   28.5     43.5 0.390   0.309   0.472
##  3 IgA         2  12.6  1.61    9.47    15.8 0.137   0.103   0.171
##  4 IgD         0 231.   4.70  222.     240.  1       0.960   1.04
##  5 IgD         1 220.   7.10  206.     234.  0.952   0.891   1.01
##  6 IgD         2 203.  11.7   180.     226.  0.878   0.779   0.978
##  7 IgG         0  95.9  5.56   85.0    107.  1       0.886   1.11
##  8 IgG         1  60.3  4.64   51.2     69.4 0.629   0.534   0.724
##  9 IgG         2  39.3  4.02   31.5     47.2 0.410    0.328    0.492
## 10 IgM         0 251.   2.67  245.     256.  1        0.979    1.02
## 11 IgM         1 247.   3.84  240.     255.  0.986    0.956    1.02
## 12 IgM         2 242.   6.07  230.     254.  0.964    0.917    1.01
```

```
# Plot results at q=0 and q=2
# Plot the mean and standard deviations at q=0 and q=2
plot(isotype_test, 0, colors=IG_COLORS, main_title=isotype_main,
     legend_title="Isotype")
```



```
plot(isotype_test, 2, colors=IG_COLORS, main_title=isotype_main,
     legend_title="Isotype")
```


Isotype diversity
Mean ²D ± SD
IgA
IgD
IgG
IgM
Isotype
● IgA
● IgD
● IgG
● IgM