Fitting species abundance models with maximum likelihood
Quick reference for `sads` package

Paulo Inácio Prado, Murilo Dantas Miranda and Andre Chalom
Theoretical Ecology Lab
LAGE at the Dep of Ecology, USP, Brazil
prado@ib.usp.br

October 13, 2024

1 Introduction

Species abundance distributions (SADs) are one of the basic patterns of ecological communities (McGill et al., 2007). The empirical distributions are traditionally modeled through probability distributions. Hence, the maximum likelihood method can be used to fit and compare competing models for SADs. The package `sads` provides functions to fit the most used models to empirical SADs. The resulting objects have methods to evaluate fits and compare competing models. The package also allows the simulation of SADs expected from communities' samples, with and without aggregation of individuals of the same species.

2 Installation

The package is available on CRAN and can be installed in **R** with the command:

```
> install.packages('sads')
```

then loaded by

```
> library(sads)
```

2.1 Developer version

The current developer version can be installed from GitHub with:

```
> library(devtools)
> install_github(repo = 'piLaboratory/sads', ref= 'dev', build_vignettes = TRUE)
```

And then load the package:

```
> library(sads)
```

3 Exploratory analyses

Throughout this document we'll use two data sets of abundances from the sads package. For more information on these data please refer to their help pages:

```
> data(moths)# William's moth data used by Fisher et al (1943)
> data(ARN82.eB.apr77)# Arntz et al. benthos data
> data(birds)# Bird census used by Preston (1948)
```

3.1 Octaves

Function `octav` tabulates the number of species in classes of logarithm of abundances at base 2 (Preston's octaves) and returns a data frame [1]:

```
> (moths.oc <- octav(moths))

Object of class "octav"
   octave upper Freq
1       0     1   35
2       1     2   11
3       2     4   29
4       3     8   32
5       4    16   26
6       5    32   32
7       6    64   31
8       7   128   13
9       8   256   19
10      9   512    5
11     10  1024    6
12     11  2048    0
13     12  4096    1
14     13  8192    0
```

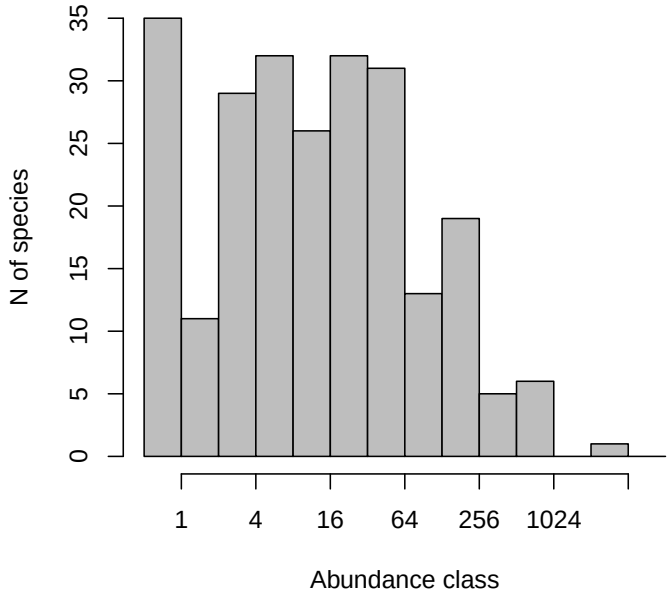
```
> (arn.oc <- octav(ARN82.eB.apr77))
```

[1] actually an object of class *octav* which inherits from class *dataframe*

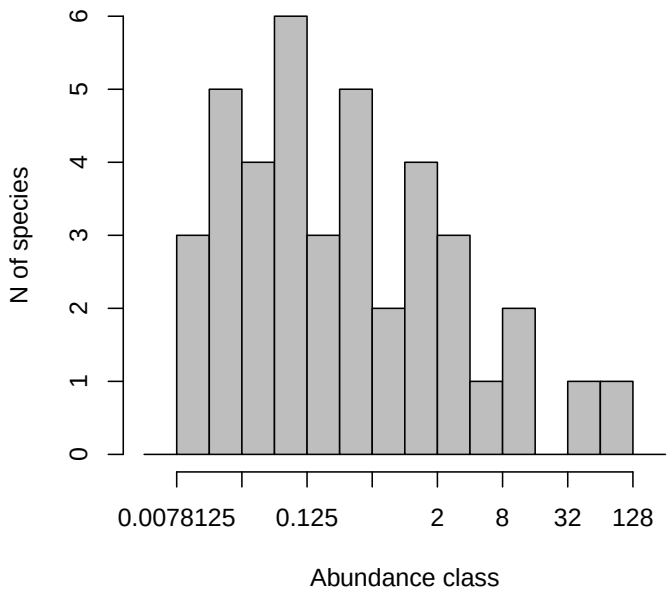
```
Object of class "octav"
   octave      upper Freq
1      -7 7.8125e-03    0
2      -6 1.5625e-02    3
3      -5 3.1250e-02    5
4      -4 6.2500e-02    4
5      -3 1.2500e-01    6
6      -2 2.5000e-01    3
7      -1 5.0000e-01    5
8       0 1.0000e+00    2
9       1 2.0000e+00    4
10      2 4.0000e+00    3
11      3 8.0000e+00    1
12      4 1.6000e+01    2
13      5 3.2000e+01    0
14      6 6.4000e+01    1
15      7 1.2800e+02    1
16      8 2.5600e+02    0
```

A logical argument `preston` allows smoothing the numbers as proposed by Preston (1948). The octave number is the upper limit of the class in log2 scale. Hence, for abundance values smaller than one (*e.g.* biomass data) the octave numbers are negative. A Preston plot is a histogram of this table, obtainable by applying the function `plot` to the data frame:

```
> plot(moths.oc)
```


N of species
0
5
10
15
20
25
30
35
1
4
16
64
256
1024
Abundance class

```
> plot(arn.oc)
```



The plot method for objects of class `octav` has a logical argument `prop` that rescales the y-axis to relative frequencies of species in each octave, which can be used to compare different data sets:

```
> plot(moths.oc, prop = TRUE, border=NA, col=NA)
> lines(octav(birds), mid = FALSE, prop = TRUE, col="red")
> lines(octav(moths), mid = FALSE, prop = TRUE)
> legend("topright", c("Preston's birds", "Fisher's moths"), col=c("red", "blue"), lty=1, bty
```

3.2 Rank-abundance plots

Function `rad` returns a data frame of sorted abundances and their ranks [2]:

```
> head(moths.rad <- rad(moths))
```

```
  rank abund
1    1  2349
2    2   823
```

[2] actually an object of class *rad* which inherits from class *dataframe*

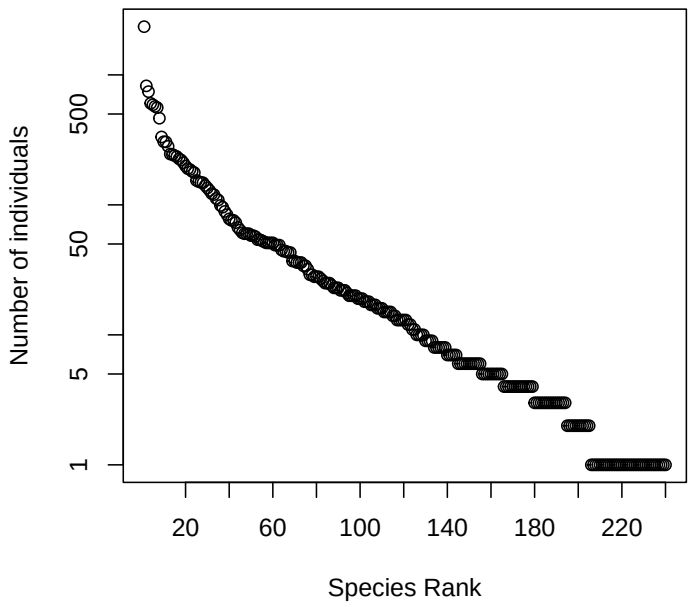
```
3    3   743
4    4   604
5    5   589
6    6   572
```

```
> head(arn.rad <- rad(ARN82.eB.apr77))
```

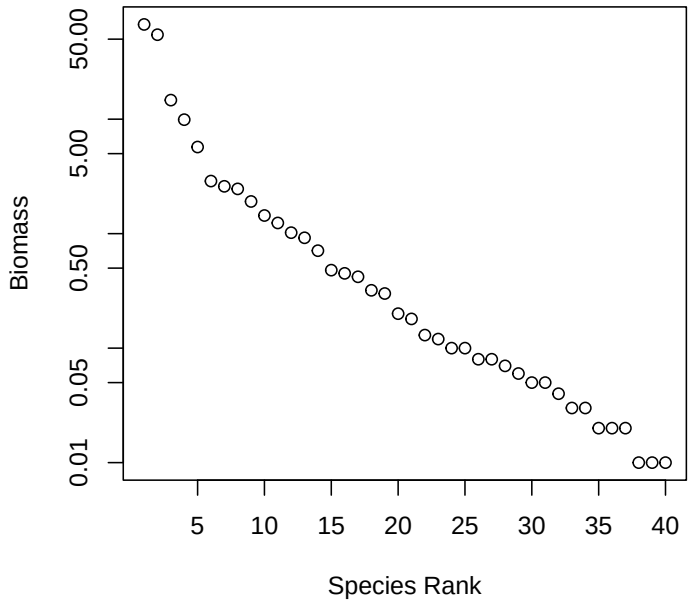
```
     rank abund
sp17    1 67.21
sp11    2 54.67
sp33    3 14.67
sp9     4  9.90
sp30    5  5.71
sp10    6  2.88
```

To get the rank-abundance or Whitaker's plot apply the function `plot` on the data frame:

```
> plot(moths.rad, ylab="Number of individuals")
```



```
> plot(arn.rad, ylab="Biomass")
```



Again, the plot method for `rad` has a logical argument `prop` rescales the y-axis to depict relative abundances:

```
> plot(moths.rad, prop = TRUE, type="n")
> lines(rad(birds), prop = TRUE, col="red")
> lines(rad(moths), prop = TRUE)
> legend("topright", c("Preston's birds", "Fisher's moths"), col=c("red", "blue"), lty=1, bty
```

4 Model fitting

The *sads* package provides maximum-likelihood fits of many probability distributions to empirical sads. The working horses are the functions `fitsad` for fitting species abundance distributions and `fitrad` for fitting rank-abundance distributions. The first argument of these functions is the vector of observed abundances [3] The second argument is the name of the model to be fitted. Please refer to the help page of the functions for details on the

[3] `fitrad` also accepts a rank-abundance table returned by function `rad` as its first argument.

models. For more information on the fitting procedure see also the vignette of the *bbmle* package, on top of which the package *sads* is built.
To fit a log-series distribution use the argument `sad='ls'`:

```
> (moths.ls <- fitsad(moths,'ls'))

Maximum likelihood estimation
Type: discrete  species abundance distribution
Species: 240 individuals: 15609

Call:
mle2(minuslogl = function (N, alpha)
-sum(dls(x, N, alpha, log = TRUE)), start = list(alpha = 40.247281791951),
    method = "Brent", fixed = list(N = 15609L), data = list(x = list(
        1, 1, 1, 1, 1, "etc")), lower = 0, upper = 240L)

Coefficients:
          N       alpha
15609.00000    40.24728

Log-likelihood: -1087.71
```

The resulting model object inherits from *mle2* (Bolker & R Development Core Team, 2014), and has all usual methods for model objects, such as summaries, log-likelihood, and AIC values:

```
> summary(moths.ls)

Maximum likelihood estimation

Call:
mle2(minuslogl = function (N, alpha)
-sum(dls(x, N, alpha, log = TRUE)), start = list(alpha = 40.247281791951),
    method = "Brent", fixed = list(N = 15609L), data = list(x = c(1L,
    1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
    1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
    1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
    3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L,
    4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 5L,
    5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 6L, 6L, 6L, 6L, 6L, 6L,
    6L, 6L, 6L, 6L, 6L, 7L, 7L, 7L, 7L, 7L, 8L, 8L, 8L, 8L, 8L,
```

```
    8L, 9L, 9L, 9L, 9L, 10L, 10L, 10L, 10L, 11L, 11L, 12L, 12L,
    13L, 13L, 13L, 13L, 13L, 14L, 14L, 15L, 15L, 15L, 15L, 16L,
    16L, 16L, 17L, 17L, 17L, 18L, 18L, 18L, 19L, 19L, 19L, 20L,
    20L, 20L, 20L, 21L, 22L, 22L, 22L, 23L, 23L, 23L, 24L, 25L,
    25L, 25L, 26L, 27L, 28L, 28L, 28L, 29L, 29L, 32L, 34L, 34L,
    36L, 36L, 36L, 37L, 37L, 43L, 43L, 44L, 44L, 45L, 49L, 49L,
    49L, 51L, 51L, 51L, 51L, 52L, 53L, 54L, 54L, 57L, 58L, 58L,
    60L, 60L, 60L, 61L, 64L, 67L, 73L, 76L, 76L, 78L, 84L, 89L,
    96L, 99L, 109L, 112L, 120L, 122L, 129L, 135L, 141L, 148L,
    149L, 151L, 154L, 177L, 181L, 187L, 190L, 199L, 211L, 221L,
    226L, 235L, 239L, 244L, 246L, 282L, 305L, 306L, 333L, 464L,
    560L, 572L, 589L, 604L, 743L, 823L, 2349L)), lower = 0, upper = 240L)

Coefficients:
      Estimate Std. Error z value     Pr(z)
alpha   40.247      6.961  5.7818 7.391e-09 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Fixed parameters:
    N
15609

-2 log L: 2175.425
```

```
> coef(moths.ls)
```

```
          N       alpha
15609.00000    40.24728
```

```
> logLik(moths.ls)
```

```
'log Lik.' -1087.713 (df=1)
```

```
> AIC(moths.ls)
```

```
[1] 2177.425
```

On the above examples, notice that the `print` method[4] displays some statistics on the input data and fitting function used - number of species, number of individuals, truncation point

[4] Or, equivalently, the `show` method

for the probability distribution (when used, see below) and whether we are fitting a discrete or continuous sad or rad - while the `summary` method displays information more associated with the fitting *per se*: standard errors and significance codes for each parameter. Also, notice that the input data is displayed by both methods, but the `print` method only shows the first values, as the complete list can be quite large.

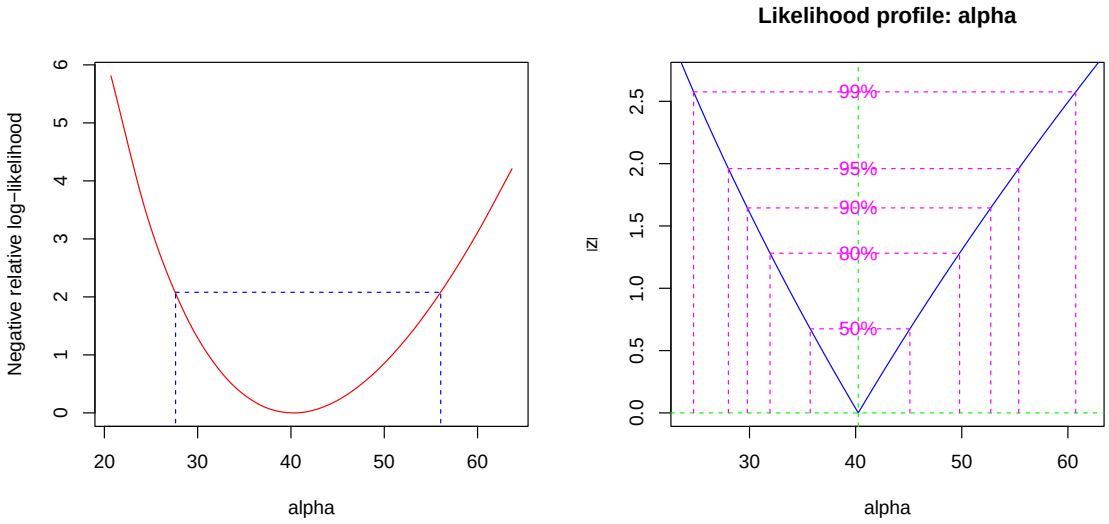
4.1 Model diagnostics

Many other diagnostic and functions are available for sad and rad models. To get likelihood profiles, likelihood intervals and confidence intervals use:

```
> moths.ls.prf <- profile(moths.ls)
> likelregions(moths.ls.prf) #likelihood intervals

Likelihood regions for ratio = 2.079442

 alpha:
        lower    upper
[1,] 27.57325 56.04053

> confint(moths.ls.prf)

   2.5 %   97.5 %
28.01537 55.36267
```

Then use `plotprofmle` to plot likelihood profiles at the original scale (relative negative log-likelihood) and function `plot` to get plots at chi-square scale (square-root of twice the relative log-likelihood):

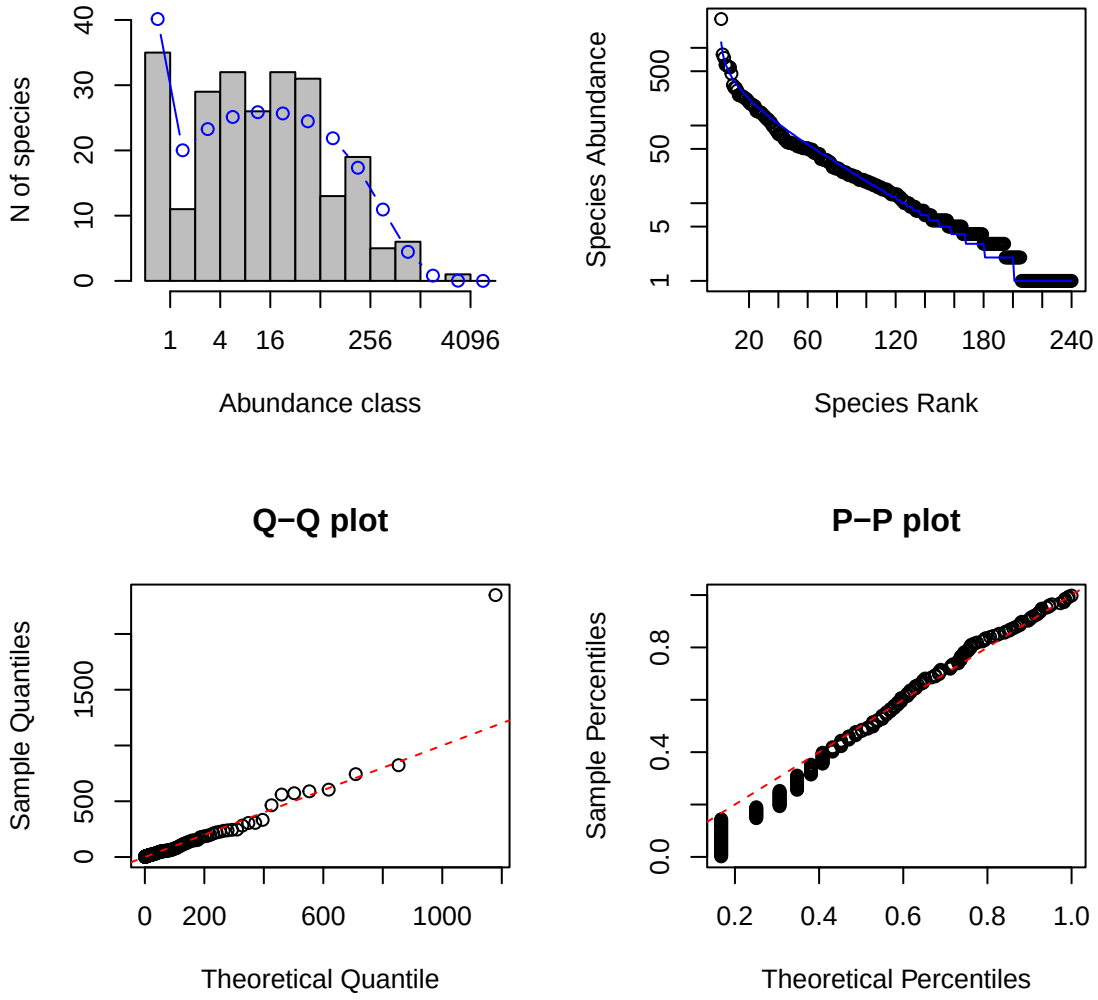
```
> par(mfrow=c(1,2))
> plotprofmle(moths.ls.prf)# log-likelihood profile
> plot(moths.ls.prf)# z-transformed profile
> par(mfrow=c(1,1))
```



Likelihood intervals and confindence intervals: Likelihood intervals include all values of the parameters that have up to a given log-likelihood absolute difference to the maximum likelihood estimate. This difference is the log-likelihood ratio and is set with the argument `ratio` of function `likelregions`. The default value of `ratio` is log(8), and thus in the example above the likelihood interval encloses all values of the parameter that are up to 8 times as plausible as the estimated value of $\alpha = 40.25$.
Likelihood intervals at log(8) converge to the value of confidence intervals at 95% as sample size increases. In most cases even for moderate sample sizes the limits of confidence and likelihood intervals are very close. Discrepancies occur only when the likelihood profile is highly asymmetric or have local *minima*. But in this kind of profile usually indicates an ill-behaved fit, and so the intervals may not be meaningful anyway.

When applied on a sad model object, the function `plot` returns four diagnostic plots:

```
> par(mfrow=c(2,2))
> plot(moths.ls)
> par(mfrow=c(1,1))
```



The first two plots (top right and left) are the octave and rank-abundance plots with the predicted values of number of species in each octave and of each species' abundance. The two last plots (bottom) are quantile-quantile and percentile-percentile graphs of the observed vs. predicted abundances. The straight line indicates the expected relation in case of perfect fit.

4.2 SADs vs RADs

Species-abundance models assign a probability for each abundance value. Thus, these models are probability density functions (PDFs) of abundances of species. Rank-abundance models assign a probability for each **abundance rank**. They are PDFs for rankings of species. The models are interchangeable (May, 1975), but currently only four rad models are available in package sads through the argument `rad` of function `fitrad`:

- "gs": geometric series (which is NOT geometric PDF, available in `fitsad` as "geom")
- "rbs": broken-stick model (MacArthur, 1957; May, 1975)
- "zipf": Zipf power-law distribution
- "mand": Zipf-Mandelbrot power-law distribution

Comparison to `radfit` from *vegan* package:
fits by `fitsad`, `fitrad` and `radfit` of *vegan* package provide similar estimates of model coefficients but not comparable likelihood values. The reason for this is the fact each function fits models that assign probability values to data in different ways. Function `fitsad` fits PDFs to observed abundances and `fitrad` fits PDFs to the ranks of the abundances. Finally, `radfit` of *vegan* fits a Poisson generalized linear model to the *expected abundances* deduced from rank-abundance relationships from the corresponding sads and rads models (Wilson, 1991). See also the help page of `radfit`. Therefore **likelihoods obtained from these three functions are not comparable**.

5 Model selection

It's possible to fit other models to the same data set, such as the Poisson-lognormal and a truncated lognormal:

```
> (moths.pl <- fitsad(x=moths, sad="poilog"))#default is zero-truncated

Maximum likelihood estimation
Type: discrete  species abundance distribution
Species: 240 individuals: 15609

Call:
mle2(minuslogl = function (mu, sig)
-sum(dtrunc("poilog", x = x, coef = list(mu = mu, sig = sig),
```

```
    trunc = trunc, log = TRUE)), start = list(mu = 1.99664912324681,
    sig = 2.18726037265132), data = list(x = list(1, 1, 1, 1,
    1, "etc")))

Coefficients:
      mu      sig
1.996463 2.187131

Truncation point: 0

Log-likelihood: -1086.07
```

```
> (moths.ln <- fitsad(x=moths, sad="lnorm", trunc=0.5)) # lognormal truncated at 0.5
```

```
Maximum likelihood estimation
Type: continuous  species abundance distribution
Species: 240 individuals: 15609

Call:
mle2(minuslogl = function (meanlog, sdlog)
-sum(dtrunc("lnorm", x, coef = list(meanlog = meanlog, sdlog = sdlog),
    trunc = trunc, log = TRUE)), start = list(meanlog = 2.57905878609957,
    sdlog = 1.78235276032689), data = list(x = list(1, 1, 1,
    1, 1, "etc")))

Coefficients:
 meanlog    sdlog
2.274346 2.039740

Truncation point: 0.5

Log-likelihood: -1086.36
```

moreover, the function `AICtab` and friends from the *bbmle* package can be used to get a model selection table:

```
> AICtab(moths.ls, moths.pl, moths.ln, base=TRUE)
```

```
         AIC    dAIC   df
moths.pl 2176.1    0.0 2
moths.ln 2176.7    0.6 2
moths.ls 2177.4    1.3 1
```

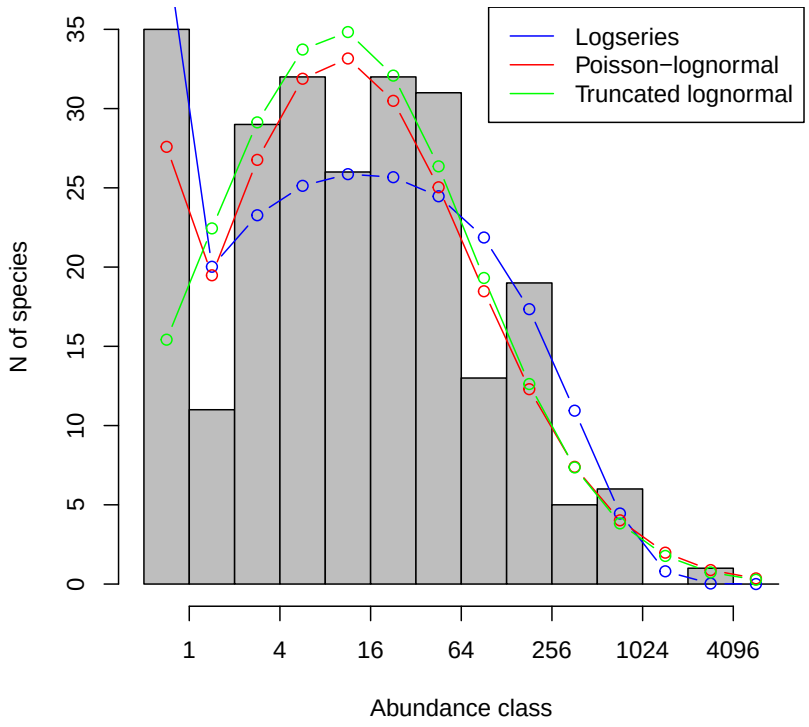
NOTICE that the information criterion methods do not differentiate between `fitsad` and `fitrad` methods. Because of this, it is possible to include `fitsad` and `fitrad` objects in the same IC-table without generating an error, but the result will be meaningless.
To compare visually fits first get octave tables:

```
> head(moths.ls.oc <- octavpred(moths.ls))

  octave upper     Freq
1      0     1 40.14377
2      1     2 20.02026
3      2     4 23.27123
4      3     8 25.12674
5      4    16 25.86285
6      5    32 25.67116

> head(moths.pl.oc <- octavpred(moths.pl))

  octave upper     Freq
1      0     1 27.58748
2      1     2 19.48221
3      2     4 26.76474
4      3     8 31.88373
5      4    16 33.16136
6      5    32 30.49056

> head(moths.ln.oc <- octavpred(moths.ln))

  octave upper     Freq
1      0     1 15.41886
2      1     2 22.44066
3      2     4 29.13034
4      3     8 33.72746
5      4    16 34.82976
6      5    32 32.08088
```

then use `lines` to superimpose the predicted values on the octave plot:

```
> plot(moths.oc)
> lines(moths.ls.oc, col="blue")
> lines(moths.pl.oc, col="red")
> lines(moths.ln.oc, col="green")
```

```
> legend("topright",
        c("Logseries", "Poisson-lognormal", "Truncated lognormal"),
        lty=1, col=c("blue","red", "green"))
```



To do the same with rank-abundance plots get the rank-abundance objects:

```
> head(moths.ls.rad <- radpred(moths.ls))
```

```
  rank abund
1    1  1180
2    2   854
3    3   710
4    4   619
5    5   554
6    6   503
```

```
> head(moths.pl.rad <- radpred(moths.pl))
```

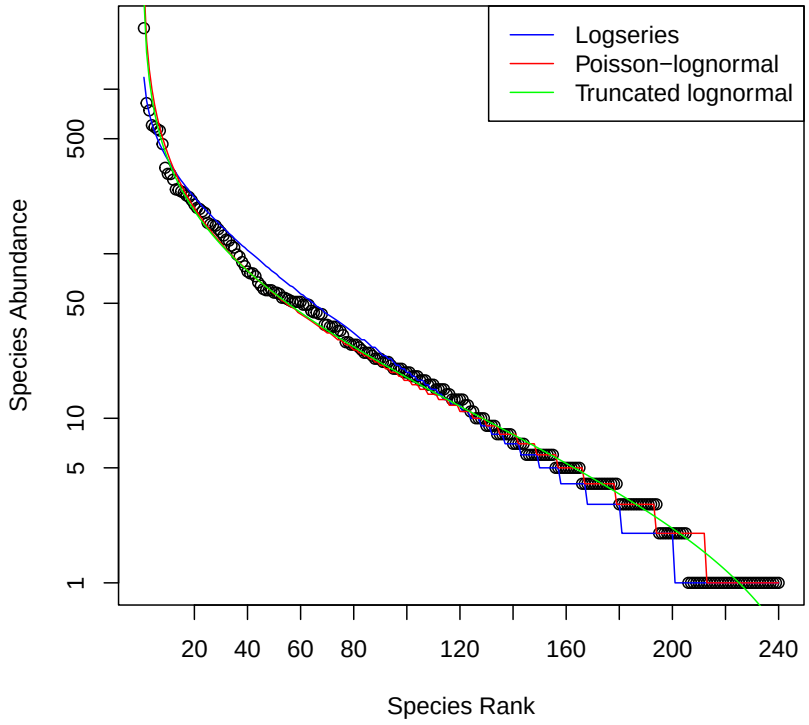
```
  rank abund
1    1  4348
2    2  1973
3    3  1322
4    4  1001
5    5   807
6    6   676
```

```
> head(moths.ln.rad <- radpred(moths.ln))
```

```
  rank     abund
1    1 3524.2394
2    2 1674.8603
3    3 1148.3539
4    4  883.6309
5    5  720.7864
6    6  609.2707
```

then plot observed and predicted values:

```
> plot(moths.rad)
> lines(moths.ls.rad, col="blue")
> lines(moths.pl.rad, col="red")
> lines(moths.ln.rad, col="green")
> legend("topright",
        c("Logseries", "Poisson-lognormal", "Truncated lognormal"),
        lty=1, col=c("blue","red", "green"))
```



5.1 Abundance class data

For ecological communities, the representation of species abundances typically occurs through categorization. For instance, when assessing the abundance of sessile organisms, it is common practice to utilize a scale based on the coverage of sampling areas, beacause direct enumeration of individuals or estimation of biomass are are extremely laborious.
The package `sads` has a specific class for fitting continuous distributions for this kind of data. We will show the use of this class using the data from de Souza Vieira & Overbeck (2020), who provide the coverage class of each plant species in plots set in grasslands in Southern Brazil. The object `grasslands` has the data from plot 'CA8', which has the largest number of species recorded in this study.

```
> head(grasslands)
```

```
                      class  cover upper mids
Andropogon lateralis    2.0 15-25%    25 20.0
```

```
Andropogon macrothrix   0.1    <1%     1  0.5
Axonopus affinis        0.1    <1%     1  0.5
Baccharis trimera       0.1    <1%     1  0.5
Briza poaemorpha        0.1    <1%     1  0.5
Briza uniolae           0.1    <1%     1  0.5
```

The vector `cover` in this data frame has the cover class for each plant species, that is, the the proportion of the area of the plot covered by all individuals of each species. Coverage is expressed in a scale with 13 intervals, as defined by the breakpoints below:

```
> (grass.brk <- c(0,1,3,5,seq(15,100, by=10),100) )

 [1]   0   1   3   5  15  25  35  45  55  65  75  85  95 100
```

We can tally the number of species in each cover class using a histogram:

```
>  grass.h <- hist(grasslands$mids, breaks = grass.brk, plot = FALSE)
```

The resulting object of the class `histogram` has the number of species in each cover class, as well as the classes midpoints:

```
>  data.frame(midpoint = grass.h$mids, N.spp = grass.h$counts)

   midpoint N.spp
1       0.5    16
2       2.0     9
3       4.0     1
4      10.0     1
5      20.0     1
6      30.0     1
7      40.0     0
8      50.0     0
9      60.0     0
10     70.0     0
11     80.0     0
12     90.0     0
13     97.5     0
```

The function `fitsadC` fits continuous distributions usually applied to describe this kind of data. The commands below fit all models currently available for abundance class data in the package `sads`. Note that the data is provided to this function through an object of the class `histogram`:

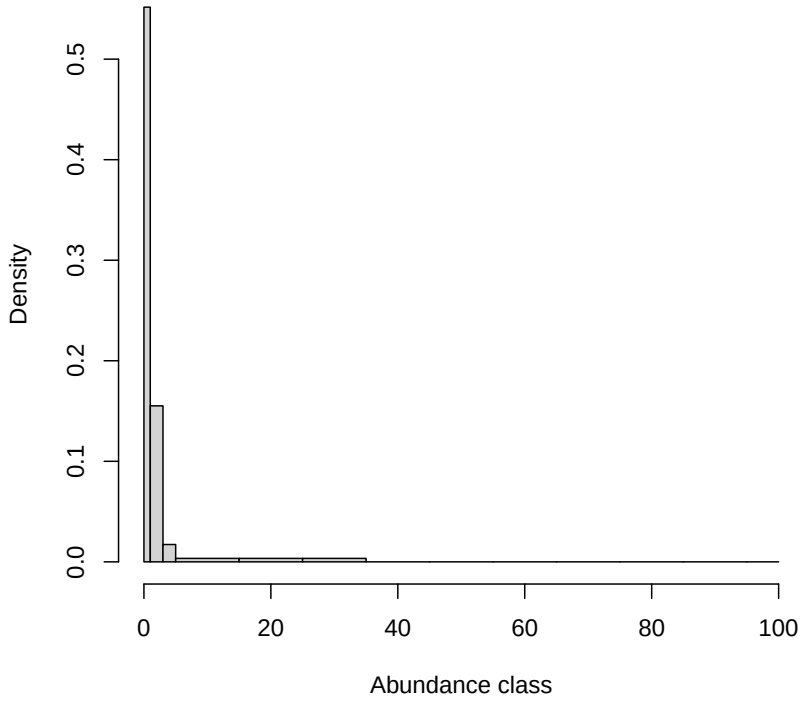
```
> grass.e <- fitsadC(grass.h, 'exp') # Exponential
> grass.g <- fitsadC(grass.h, 'gamma') # Pareto
> grass.l <- fitsadC(grass.h, 'lnorm') # Log-normal
> grass.p <- fitsadC(grass.h, 'pareto') # Pareto
> grass.w <- fitsadC(grass.h, 'weibull') # Weibull
```

The fitted models are of class `fitsadC`, for which most of the methods for diagnostics and model comparison showed in the previous sections are available. For instance, the fitted models can be compared in the usual way:

```
> AICctab(grass.e, grass.g, grass.l,
          grass.p, grass.w,
          weights = TRUE, base = TRUE)

        AICc dAICc df weight
grass.p 76.6  0.0  2  0.487
grass.l 77.6  1.0  2  0.291
grass.w 78.9  2.3  2  0.155
grass.g 80.6  4.0  2  0.067
grass.e 95.2 18.6  1  <0.001
```

A histogram with observed values can ploted simply with

```
> plot(grass.h, main = "", xlab = "Abundance class")
```



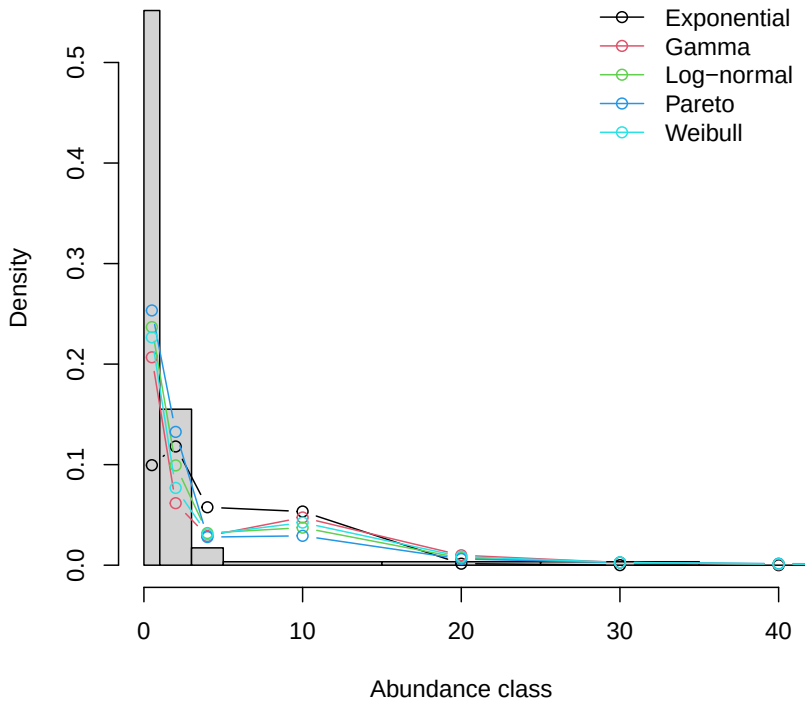
By default, R plots histograms with classes of unequal size using a density scale. The function `coverpred` provides the number of species expected by a fiited model, in frequency, relative frequency and density scales.

```
> ## Predicted by each model
> grass.e.p <- coverpred(grass.e)
> grass.g.p <- coverpred(grass.g)
> grass.l.p <- coverpred(grass.l)
> grass.p.p <- coverpred(grass.p)
> grass.w.p <- coverpred(grass.w)
```

We can then use this object to add the densities values predicted by each model:

```
> ## Plot
> plot(grass.h, main = "", xlab = "Abundance class", xlim = c(0,40))
> ## Adds predicted points
```

```
> points(grass.e.p, col = 1)
> points(grass.g.p, col = 2)
> points(grass.l.p, col = 3)
> points(grass.p.p, col = 4)
> points(grass.w.p, col = 5)
> legend("topright",
        legend = c("Exponential", "Gamma", "Log-normal", "Pareto", "Weibull"),
        col = 1:5, bty = "n",
        lty = 1 , pch = 1)
```



Please refer to the man pages of `fitsadC` and `fitsaC-class` for further methods and usage.

6 Simulations

The function `rsad` returns random samples of a community with S species. The mean abundances of the species in the communities are independent identically distributed (*iid*)

variables that follow a given probability distribution. The sample simulates a given number of draws of a fraction a from the total number of individuals in the community. For instance, to simulate two Poisson samples of 10% of a community with 10 species that follows a lognormal distribution with parameters $\mu = 3$ and $\sigma = 1.5$ use:

```
> set.seed(42)# fix random seed to make example reproducible
> (samp1 <- rsad(S = 10, frac = 0.1, sad = "lnorm",
                 coef=list(meanlog = 3, sdlog = 1.5),
                 zeroes=TRUE, ssize = 2))

   sample species abundance
1       1       1        20
2       1       2         4
3       1       3         7
4       1       4         2
5       1       5         4
6       1       6         1
7       1       7        25
8       1       8         3
9       1       9        45
10      1      10         1
11      2       1        17
12      2       2         2
13      2       3         0
14      2       4         3
15      2       5         6
16      2       6         2
17      2       7        18
18      2       8         0
19      2       9        53
20      2      10         4
```

The function returns a data frame with a sample numeric label, species' numeric label and species' abundance in each sample. By default, `rsad` returns a vector of abundances of single Poisson sample with zeroes omitted:

```
> (samp2 <- rsad(S = 100, frac=0.1, sad="lnorm",
                 list(meanlog=5, sdlog=2)))

 [1]  155  697    4    7   48    5   40   56  105    8   48
[12]    1    3    1   14   21    6   66    2    3   32  259
```

```
[23]    8   51   21    1  312   42   23   20   48   12   28
[34]   14   20   40  267    5  209   36  107   93   58    1
[45]    7   39    2    7   56   70   31    3    4  305   25
[56]   15   12    3   48    8   12  101   69  255    5   51
[67]  253    4    1    2   17   49  187  121  599    3   23
[78]   12    9   16   21   10   17    3    5    2    9    5
[89] 3214    1   19    1   31
```

Since this is a Poisson sample of a lognormal community, the abundances in the sample should follow a Poisson-lognormal distribution with parameters $\mu + \log a$ and σ (Grøtan & Engen, 2008). We can check this by fitting a Poisson-lognormal model to the sample:

```
> (samp2.pl <- fitsad(samp2, "poilog"))

Maximum likelihood estimation
Type: discrete  species abundance distribution
Species: 93 individuals: 8759

Call:
mle2(minuslogl = function (mu, sig)
-sum(dtrunc("poilog", x = x, coef = list(mu = mu, sig = sig),
    trunc = trunc, log = TRUE)), start = list(mu = 2.70913763997913,
    sig = 1.88422112870725), data = list(x = list(155, 697, 4,
    7, 48, "etc")))

Coefficients:
      mu      sig
2.709138 1.884220

Truncation point: 0

Log-likelihood: -453.22

> ## checking correspondence of parameter mu
> coef(samp2.pl)[1] - log(0.1)

      mu
5.011723
```

Not bad. By repeating the sampling and the fit many times it's possible to evaluate the bias and variance of the maximum likelihood estimates:

```
> results <- matrix(nrow=75,ncol=2)
> for(i in 1:75){
     x <- rsad(S = 100, frac=0.1, sad="lnorm",
               list(meanlog=5, sdlog=2))
     y <- fitsad(x, "poilog")
     results[i,] <- coef(y)
 }
> results[,1] <- results[,1]-log(0.1)
```

Bias is estimated as the difference between the mean of estimates and the value of parameters:

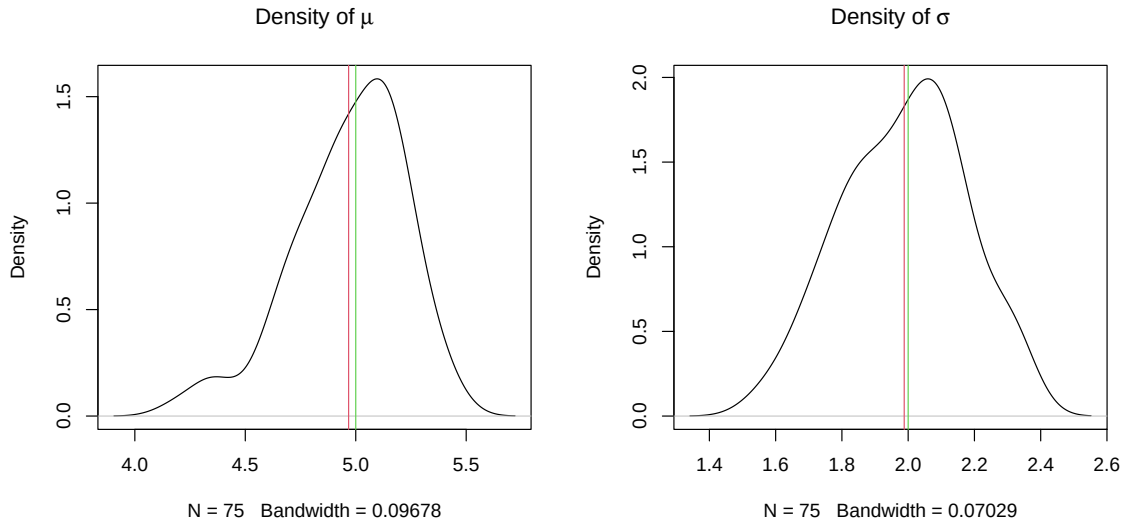
```
> ##Mean of estimates
> apply(results,2,mean)

[1] 4.967704 1.988043

> ## relative bias
> (c(5,2)-apply(results,2,mean))/c(5,2)

[1] 0.006459158 0.005978719
```

And the precision of the estimates are their standard deviations

```
> ##Mean of estimates
> apply(results,2,sd)

[1] 0.2550191 0.1852096

> ## relative precision
> apply(results,2,sd)/apply(results,2,mean)

[1] 0.05133541 0.09316177
```

Finally, a density plot with lines indicating the mean of estimates and the values of parameters:

```
> par(mfrow=c(1,2))
> plot(density(results[,1]), main=expression(paste("Density of ",mu)))
> abline(v=c(mean(results[,1]),5), col=2:3)
> plot(density(results[,2]), main=expression(paste("Density of ",sigma)))
> abline(v=c(mean(results[,2]), 2), col=2:3)
> par(mfrow=c(1,1))
```



Increasing the number of simulations improves these estimators.

7 Bugs and issues

The package project is hosted on GitHub (`https://github.com/piLaboratory/sads/`). Please report bugs and issues and give us your feedback at `https://github.com/piLaboratory/sads/issues`.

References

Bolker, B. & R Development Core Team, 2014. bbmle: Tools for general maximum likelihood estimation. R package version 1.0.16.

de Souza Vieira, M. & G. E. Overbeck, 2020. Small seed bank in grasslands and tree plantations in former grassland sites in the south brazilian highlands. *Biotropica* **52**:775–782.

Grøtan, V. & S. Engen, 2008. poilog: Poisson lognormal and bivariate Poisson lognormal distribution. R package version 0.4.

MacArthur, R., 1957. On the relative abundance of bird species. *Proceedings of the National Academy of Sciences of the United States of America* **43**:293.

May, R. M., 1975. Patterns of species abundance and diversity. In M. L. Cody & J. M. Diamond, editors, *Ecology and Evolution of Communities*, chapter 4, pages 81–120. Harvard University Press, Cambridge, MA.

McGill, B., R. Etienne, J. Gray, D. Alonso, M. Anderson, H. Benecha, M. Dornelas, B. Enquist, J. Green, F. He, A. Hurlbert, A. E. Magurran, P. Marquet, B. Maurer, A. Ostling, C. Soykan, K. Ugland, & E. White, 2007. Species abundance distributions: moving beyond single prediction theories to integration within an ecological framework. *Ecology Letters* **10**:995–1015.

Preston, F. W., 1948. The commonness and rarity of species. *Ecology* **29**:254–283.

Wilson, J., 1991. Methods for fitting dominance/diversity curves. *Journal of Vegetation Science* **2**:35–46.