MIDAS package vignette

1 Overview

The MIDAS package (v0.1.0) implements the **MI**crobiome **DA**ta **S**imulator functions that simulate realistic microbiome data, maintaining the distributional and taxon-taxon correlation of a template microbiome datase. Users may also change the parameter setup in the model to introduce "effect' ' to facilitate simulation.

1.1 Required packages

Required packages for functions in MIDAS include: psych, MASS, vegan, pracma, scam. These packages are all available on CRAN.

1.2 Installation

Install from the local file (the .tar.gz file), which you can download from https://github.com/mengyu-he/MIDAS.

```
devtools::install_local("MIDAS_0.1.0.tar.gz", dependencies = TRUE)
```

A github installation is also available via

```
install_github("mengyu-he/MIDAS")
```

1.3 Open the Vignette in R

```
browseVignettes("MIDAS")
```

or

```
vignette("MIDAS_vignette", package = "MIDAS")
```

2 MIDAS: Microbiome Data Simulator

There are three main functions in MIDAS package: *Midas.setup()*, *Midas.modify()*, *Midas.sim()*.

2.1 Template data description

We demonstrate the method of MIDAS and functions in this package using a filtered microbiome dataset of patients with IBD(Inflammatory Bowel Disease) in Human Microbiome Project 2 (HMP2) [1]. A filtered version of the data is included in the MIDAS package and can be directly loaded through

```
data("count.ibd")
```

This "count.ibd" were a modified version of the IBD data in package **HMP2Data** on Bioconductor by filtering out samples with library sizes smaller than 3000 and taxa that appear in less than 2 samples. The filtered data have 146 rows (samples) and 614 columns (taxa).

```
library(HMP2Data)
IBD16S()
```

2.2 Template data description

We demonstrate the method of MIDAS and functions in this package using a filtered microbiome dataset of patients with IBD(Inflammatory Bowel Disease) in Human Microbiome Project 2 (HMP2) [1]. The data can be directly loaded through

```
data("count.ibd")
```

This HMP2 data were modified through package **HMP2Data** on Bioconductor by filtering out samples with library sizes smaller than 3000 and taxa that appear in less than 2 samples. The filtered data have 146 rows (samples) and 614 columns (taxa).

```
library(HMP2Data)
IBD16S()

count.ibd <- t(IBD16S_mtx[, colSums(IBD16S_mtx)>3000] )
count.ibd <- count.ibd[, colSums(count.ibd>0)>1]

dim(count.ibd)
```

Other datasets come with the package include a vaginal microbiome dataset from HMP2 [1] and a upper-respiratory-tract microbiome dataset [2].

2.3 Setup the MIDAS model

The function *Midas.setup()* fits the underlying MIDAS model and extracts information from the template data and returns the estimated parameters for microbiome simulation. To fit MIDAS to the count.ibd data, the following command can be used:

```
count.ibd.setup <- Midas.setup(count.ibd, n.break.ties = 100, fit.beta = F)
```

The argument `n.break.ties` specifies he number of replicates used to break ties when ranking relative abundances for each taxon. When `fit.beta = FALSE` (default), a rank based approach is used to obtain the relative abundances for taxa that are considered to be non-zero in the simulated data. When `fit.beta = TRUE`, a beta distribution is fitted for the relative abundance of each taxon by matching the first two

moments. In particular, for taxon j, if the mean and sample variance of relative abundances $\boldsymbol{\pi}_j$ are $\bar{\boldsymbol{\pi}}_j$ and s_j^2, then the corresponding parameters of the beta distribution are

$$\alpha_j = \bar{\boldsymbol{\pi}}_j \left(\frac{\bar{\boldsymbol{\pi}}_j(1-\bar{\boldsymbol{\pi}}_j)}{s_j^2} - 1 \right),$$

$$\beta_j = \alpha_j \left(\frac{1-\bar{\boldsymbol{\pi}}_j}{\bar{\boldsymbol{\pi}}_j} \right).$$

The returned values of *Midas.setup()* include: the estimated tetrachoric correlation of observed presence-absence data `tetra.corr`, the estimated Pearson correlation of observed relative abundances `corr.rel.corrected`, the proportion of non-zero cells for each taxon `taxa.1.prop` and number of non-zero cells for each subject `num.1`, the observed mean relative abundances of each taxon `mean.rel.abund`, and the observed relative abundances among non-zero cells for each taxon `mean.rel.abund.1`.

2.4 Modify the fitted MIDAS model

The function *Midas.modify()* sets up quantities required to simulate data using MIDAS. This function also allows users to modify taxon relative abundances, library sizes, taxon proportion of non-zero cells and number of samples. Note that this is a **required** step even if no adjustment is made.

MIDAS supports two kinds of simulation strategies. In the first, users can change one set of parameters (e.g., library sizes) or even library sizes and taxon relative abundances and let MIDAS choose the remaning parameters using a generalized additive model. This strategy is designed to retain relationships between parameters found in the target data. The second strategy allows the user to specify all the parameters. For example, library size can be increased without changing the proportion of empty cells. Although this strategy may produce data that is not as realistic, it may appeal to people who are developing methods for analyzing microbiome data.

In addition to changes in library size and the number of samples, Midas.modify() allows users to directly or indirectly specify three types of quantities:

- $\hat{p}_j$: mean relative abundances of taxa (`mean.rel.abund`)
- $\hat{p}_j^1$: mean relative abundances of taxa among non-zero samples (`mean.rel.abund.1`)
- $\hat{\delta}_j$: proportion of non-zero cells for taxa (`taxa.1.prop`)

These three quantities are related by

$$\hat{p}_j = \hat{\delta}_j \hat{p}_j^1,$$

and the function will stop and give a message if all three quantities are specified by the user but the equation is not met. Other constraints include,

$$\hat{p}_j \leq \hat{p}_j^1$$
$$\hat{p}_j \leq \hat{\delta}_j$$

and failure to satisfy these constraints will also cause MIDAS to terminate.

2.4.1 No modification at all

When no additional adjustment is wanted (`lib.size = NULL`, `n.sample = NULL`, `mean.rel.abund = NULL`, `taxa.1.prop = NULL`), the number of samples and their target library sizes for the simulated data are the same as that of the template. The following code is an example.

```
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   lib.size = NULL,
                                   mean.rel.abund = NULL,
                                   taxa.1.prop = NULL)
```

We next describe how to modify the features of the data, to allow for a wide variety of simulation scenarios.

2.4.2 Modify the number of samples and their library sizes

To change the library sizes, set the lib.size argument equal to a vector with the target library sizes. The length of this vector becomes the new number of samples.

Because larger library sizes are typically accompanied by fewer zero cells, MIDAS.modify can fit a Shape Constrained Additive model (SCAM) model

$$log10(Z_{i.}) = f\left(log_{10}(N_i)\right) + \epsilon_i$$

to determine the appropriate number of zero cells for the new library size, where $Z_{ij} = 0$ if the i^{th} sample has no counts of taxon j and $Z_{ij} = 1$ otherwise.

For example, to generate data having library sizes that are uniformly distributed between 1,000 and 10,000 while changing the number of samples to 700 and allowing the proportion of non-zero cells to adjust according the the SCAM model, we execute the code

```
new.lib.size=sample(1000:10000, size=700, replace=T)
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   lib.size = new.lib.size)
```

If desired, MIDAS can retain the proportion of zero cells in each taxon as found in the target data by specifying any two taxon-level parameters, e.g. `taxa.1.prop='same'` and `mean.rel.abund='same'`. In this case, the SCAM model is not fit. Note that, specifying taxa features to be `"same"` is different from giving a `NULL` value. In the previous example, to change library sizes without changing the proportion of non-zero cells in each taxon, we use the code

```
new.lib.size=sample(1000:10000, size=146, replace=T)
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   lib.size = new.lib.size,
                                   taxa.1.prop = 'same',
                                   mean.rel.abund='same')
```

This example is exactly equivalent to scaling the rows of the count table by the new library sizes.

2.4.3 Modify features of taxa

In this section, we assume that only taxon quantities are to be modified; in section 2.4.4 we consider simultaneous modification of taxon quantities and library sizes.

2.4.3.1 Only one feature of taxa is given If only one of the above three quantities is specified by the user, then MIDAS will use a flexible model to infer appropriate values of the other two quantities. When only the mean relative abundances of taxa p is given, MIDAS.modify fits a SCAM model for δ on p,

$$logit\left(\delta_j\right) = g\left(log_{10}(p_j)\right) + \epsilon_j$$

where g is a monotone smoothing spline, δ_j is the observed proportion of non-zero cells of j-th taxon, and p_j is the observed mean relative abundances of j-th taxon. The values of p_j^1 are then determined by the constraint.

When only the proportion of non-zero cells of taxa $\hat{\delta}$ is given, MIDAS.modify fits a SCAM model for p on δ,

$$logit(p_j) = g\left(log_{10}(\delta_j)\right) + \epsilon_j$$

Note that only specifying the mean relative abundances of taxa among non-zero samples $\hat{p}^1$ is not allowed.

We next give examples of these kinds of modifications, starting with modifications that only specify a new value for the relative abundances p_j. Suppose we want to increase $log(p_j)$ by $\beta = 0.1$ for the first 10 taxa (e.g. modify these taxa according to a compositional model). The following is the code for generating a set of target relative abundances as described,

```
beta=0.1
new.mean.rel.abund <- count.ibd.setup$mean.rel.abund
new.mean.rel.abund[1:10]=exp(beta)*new.mean.rel.abund[1:10]
new.mean.rel.abund=new.mean.rel.abund/sum(new.mean.rel.abund)
```

then the target taxa relative abundances can be used in Midas.modify() as the following,

```
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   mean.rel.abund=new.mean.rel.abund)
```

As a second example, suppose the target proportions of non-zero cells for the first 10 taxa are to be increased by a factor of exp(beta) while all others are unchanged. Then, the following code can be used:

```
new.taxa.1.prop=count.ibd.setup$taxa.1.prop
new.taxa.1.prop[1:10]=pmin(exp(beta)*new.taxa.1.prop[1:10],1)
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   taxa.1.prop=new.taxa.1.prop)
```

2.4.3.2 Full information of taxa features is given

If all three quantities $\hat{p}, \hat{p}^1, \hat{\delta}$, or any two of them are given, full information of taxa features is given, then no additional model for changing taxa features is needed.

The following code is one example that specifies the relative abundances `new.mean.rel.abund` as calculated in the section 2.4.3.1 but uses the same proportion of non-zeros as in the template data.

```
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   mean.rel.abund = new.mean.rel.abund,
                                   taxa.1.prop = "same")
```

Note that, specifying taxa features to be `"same"` is different from giving a `NULL` value. If in the above example, we use `taxa.1.prop = NULL`, then the proportion of non-zeros will be changed by fitting a SCAM model and making predictions; however, if we use `taxa.1.prop = "same"`, the proportions of non-zeros of taxa will remain the same as in the template data, and is equivalent to using `taxa.1.prop = count.ibd.setup$taxa.1.prop`.

Although fully specifying the taxa features may result in users generating data that is 'less realistic' in some sense, it allows the user greater control over what parameters will be used to simulate data. This level of control is important, for example, when conducting simulation studies of programs that test for association between taxa and traits.

2.4.3.3 Adjustment for parameters of beta distribution If we set `fit.beta = TRUE` in Midas.setup(), which asks for simulating relative abundances from a beta distribution, Midas.modify() will make corresponding adjustment to the estimated shape parameters of the beta distribution. Suppose the observed relative abundances for j-th taxon are $\pi_{\cdot j}$, then the adjusted relative abundances are obtained by $\hat{\pi}_{\cdot j} = \pi_{\cdot j}^{\alpha}$, where α is chosen so that, after adjustment, the mean relative abundance in taxon j is `new.mean.rel.abund[j]` in the code example in section 2.4.3.2.

2.4.4 Modify library sizes and features of taxa simultaneously

Library sizes and taxa features can be manipulated simultaneously. As before, two strategies are supported: one in which new library sizes and taxon relative abundances are specified and MIDAS uses a SCAM model to fit the others, and one in which the uses specifies all quantities.

For example, in the following code, the library sizes and number of samples is changed to the value new.lib.size defined in section 2.4.2, and the relative abundance vector is changed to the value new.mean.rel.abund defined in section 2.4.3.1.

```
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   lib.size = new.lib.size,
                                   mean.rel.abund  = new.mean.rel.abund)
```

Because only mean.rel.abund is specified, MIDAS uses the SCAM model

$$log_{10}(Z_{i\cdot}) = f\left(log_{10}(N_i)\right) + g\left(log_{10}(p_j)\right) + \epsilon_{ij}$$

Note that MIDAS does not currently support specifying *only* a new library size and a new value of `taxa.1.prop` or *only* a new library size and `mean.rel.abund.1`. To make changes like these, the full taxon information must be specified, as described next.

Users can also fully specify the new taxon-level quantities and library sizes as described previously. For example, the following code will change the relative abundances and library sizes while keeping the same proportion of nonzero cells in each taxon as in the target data:

```
count.ibd.modified <- Midas.modify(count.ibd.setup,
                                   lib.size = new.lib.size,
                                   mean.rel.abund  = new.mean.rel.abund,
                                   taxa.1.prop = 'same'
                                   )
```

In this case, MIDAS first modifies the library size as described in section 2.4.2 and then modifies the taxon-level quantities as described in section 2.4.3.2.

2.4.5 Simulate microbiome data with the fitted MIDAS model

The function *Midas.sim()* takes the quantities obtained from Midas.modify and simulates microbiome data. If the input data was a count table, then the output is comprised of `s01_res`, the simulated 0/1 (presence-absence) data, `s_rel`, the table of relative abundances, and `sim_count`, the table of count data. If the original input data was a relative abundance table, only a 0/1 and table of relative abundances is generated.

```
simulated.data <- Midas.sim(count.ibd.modified)
summary(simulated.data)
```

2.5 References

[1] Lloyd-Price, J., Arze, C., Ananthakrishnan, A.N. *et al.* Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature* 569, 655–662 (2019). https://doi.org/10.1038/s41586-019-1237-9.

[2] Charlson, E.S., Chen, J., Custers-Allen, R. *et al.* Disordered microbial communities in the upper respiratory tract of cigarette smokers. *PloS one*, 5(12), e15216 (2010). https://doi.org/10.1371/journal.pone.0015216