Package 'sparseDOSSA'

April 11, 2023

Type Package

Title Sparse Data Observations for Simulating Synthetic Abundance

Version 1.22.0

Date 2016-10-28

Author Boyu Ren<bor158@mail.harvard.edu>, Emma Schwager<emma.schwager@gmail.com>, Timothy Tickle<ttickle@hsph.harvard.edu>, Curtis Huttenhower <chuttenh@hsph.harvard.edu>

Maintainer Boyu Ren<bor158@mail.harvard.edu>, Emma Schwager <emma.schwager@gmail.com>, George Weingart<george.weingart@gmail.com>

Description The package is to provide a model based Bayesian method to characterize and simulate microbiome data. sparseDOSSA's model captures the marginal distribution of each microbial feature as a truncated, zero-inflated log-normal distribution, with parameters distributed as a parent log-normal distribution. The model can be effectively fit to reference microbial datasets in order to parameterize their microbes and communities, or to simulate synthetic datasets of similar population structure. Most importantly, it allows users to include both known feature-feature and feature-metadata correlation structures and thus provides a gold standard to enable benchmarking of statistical methods for metagenomic data analysis.

License MIT + file LICENSE

VignetteBuilder knitr

Suggests knitr, BiocStyle, BiocGenerics, rmarkdown

Imports stats, utils, optparse, MASS, tmvtnorm (>= 1.4.10), MCMCpack

biocViews ImmunoOncology, Bayesian, Microbiome, Metagenomics, Software

RoxygenNote 5.0.1

git_url https://git.bioconductor.org/packages/sparseDOSSA

git_branch RELEASE_3_16

git_last_commit 70eea3e
git_last_commit_date 2022-11-01
Date/Publication 2023-04-10

R topics documented:

`sparseDOSSA` *Sparse Data Observations for Simulating Synthetic Abundance*

Description

Sparse Data Observations for Simulating Synthetic Abundance

Usage

```
sparseDOSSA( strNormalizedFileName = "SyntheticMicrobiome.pcl",
             strCountFileName = "SyntheticMicrobiome-Counts.pcl",
             parameter_filename = "SyntheticMicrobiomeParameterFile.txt",
             bugs_to_spike = 0,
             spikeFile = NA,
             calibrate = NA,
             datasetCount = 1,
             read_depth = 8030,
             number_features = 300,
             bugBugCorr =  "0.5",
             spikeCount = "1",
             lefse_file = NULL,
             percent_spiked = 0.03,
             minLevelPercent = 0.1,
             number_samples = 50,
             max_percent_outliers = 0.05,
             number_metadata = 5,
             spikeStrength = "1.0",
             seed =  NA,
             percent_outlier_spikins = 0.05,
             minOccurence =  0,
             verbose =  TRUE,
             minSample =  0,
             scalePercentZeros = 1,
             association_type =  "linear",
             noZeroInflate =  FALSE,
             noRunMetadata = FALSE,
             runBugBug =  FALSE )
```

Arguments

`strNormalizedFileName`
: This output file records the synthetic microbiome data for null community (no spike-in and outliers), outlier-added community without spike-in and final spiked data. We put samples in columns and features in rows. The first chunk of the file is metadata, with row names Metadata_. The second chunk is for null community, with row names Feature_Lognormal_. The third chunk is for outlier-introduced community, with row names Feature_Outlier_*. The last chunk is for spiked data, with row names Feature_spike. This file records relative abundance data.

`strCountFileName`
: This output file has the same organization as the file strNormalizedFileName but records raw counts data.

`parameter_filename`
: This output file records diagnostic information and values of model parameters as well as the spike-in assignment. The most part of this file is used only for debugging. Users can focus on lines after Minimum Spiked-in Samples. Those lines record which metadata are correlated with which feature. The format is all metadata that are correlated with a specific features are listed under the name of the feature.

`bugs_to_spike` Number of bugs to correlate with others. A non-negative integer value is expected.

`spikeFile` The name of the file where the correlation values are stored. Should have fields 'Domain', 'Range', and 'Correlation'.

`calibrate` Calibration file for generating the random log normal data. TSV file (column = feature).

`datasetCount` The number of bug-bug spiked datasets to generate. A positive integer value is expected.

`read_depth` Simulated read depth for counts. A positive integer value is expected.

`number_features`
: The number of features per sample to create. A positive integer value is expected.

`bugBugCorr` A vector of string separated values for the correlation values of the pairwise bug-bug associations. This is the correlation of the log-counts. Values are comma-separated; for example: 0.7,0.5. Default is 0.5.

`spikeCount` Counts of spiked metadata used in the spike-in dataset - These values should be comma delimited values, in the order of the spikeStrength values (if given), Can be one value, in this case the value will be repeated to pair with the spikeCount values (if multiple are present). For example 1,2,3.

`lefse_file` Folder containing lefSe inputs.

`percent_spiked` The percent of features spiked-in. A real number between 0 and 1 is expected.

`minLevelPercent`
: Minimum percent of measurements out of the total a level can have in a discontinuous metadata (rounded up to the nearest count). A real number between 0 and 1 is expected.

`number_samples` The number of samples to generate. A positive integer greater than 0 is expected.

`max_percent_outliers`
The maximum percent of outliers to spike into a sample. A real number between 0 and 1 is expected.

`number_metadata`
Indicates how many metadata are created, number_metadata*2 = number continuous metadata, number_metadata = number binary metadata, number_metadata = number quaternary metadata, A positive integer greater than 0 is expected.

`spikeStrength` Strength of the metadata association with the spiked-in feature, These values should be comma delimited and in the order of the spikeCount values (if given),Can be one value, in this case the value will be repeated to pair with the spikeStrength values (if multiple are present). For example 0.2,0.3,0.4.

`seed` A seed to freeze the random generation of counts/relative abundance,If left as default (NA), generation is random - If seeded, data generation will be random within a run but identical if ran again under the same settings,an integer is expected.

`percent_outlier_spikins`
The percent of samples to spike in outliers. A real number between 0 to 1 is expected.

`minOccurence` Minimum counts a bug can have for the occurrence quality control filter used when creating bugs (filtering minimum number of counts in a minimum number of samples). A positive integer is expected.

`verbose` If True logging and plotting is made by the underlying methodology. This is a flag, it is either included or not included in the command line, no value needed.

`minSample` Minimum samples a bug can be in for the occurrence quality control filter used when creating bugs (filtering minimum number of counts in a minimum number of samples). A positive integer is expected.

`scalePercentZeros`
A scale used to multiply the percent zeros of all features across the sample after it is derived from the relationships with it and the feature abundance or calibration file. Requires a number greater than 0. A number greater than 1 increases sparsity, a number less than 1 decreases sparsity, O removes sparsity, 1 (default) does not change the value and the value.

`association_type`
The type of association to generate. Options are 'linear' or 'rounded_linear'.

`noZeroInflate` If given, zero inflation is not used when generating a feature. This is a flag, it is either included or not included in the command line, no value needed.

`noRunMetadata` If given, no metadata files are generated, This is a flag, it is either included or not included in the command line, no value needed.

`runBugBug` If given, bug-bug interaction files are generated in addition to any metadata files. This is a flag, it is either included or not included in the command line, no value needed.

Value

A list contains the names of the output files.

Author(s)

Boyu Ren<bor158@mail.harvard.edu>, Emma Schwager<eschwager@hsph.harvard.edu>, Timothy Tickle<ttickle@hsph.harvard.edu>, Curtis Huttenhower <chuttenh@hsph.harvard.edu>

Examples

```
sparseDOSSA(strNormalizedFileName = "SyntheticMicrobiome.pcl",
strCountFileName = "SyntheticMicrobiome-Counts.pcl",
parameter_filename = "SyntheticMicrobiomeParameterFile.txt",
bugs_to_spike = 0,
calibrate = NA,
datasetCount = 1,
read_depth = 8030,
number_features = 300,
spikeCount = "1",
lefse_file = NA,
percent_spiked = 0.03,
minLevelPercent =  0.1,
number_samples = 50,
max_percent_outliers = 0.05,
number_metadata = 5,
spikeStrength =  "1.0",
seed =  1,
percent_outlier_spikins = 0.05,
minOccurence =  0,
verbose =  TRUE,
minSample =  0,
association_type =  "linear",
noZeroInflate =  FALSE,
noRunMetadata = FALSE,
runBugBug =  FALSE)
```

Index