Package: BRACoD.R (via r-universe)

September 1, 2024

Title BRACoD: Bayesian Regression Analysis of Compositional Data

Version 0.0.2.0

Description The goal of this method is to identify associations between bacteria and an environmental variable in 16S or other compositional data. The environmental variable is any variable which is measure for each microbiome sample, for example, a butyrate measurement paired with every sample in the data. Microbiome data is compositional, meaning that the total abundance of each sample sums to 1, and this introduces severe statistical distortions. This method takes a Bayesian approach to correcting for these statistical distortions, in which the total abundance is treated as an unknown variable. This package runs the python implementation using reticulate.

Imports reticulate

Config/reticulate list(packages = list(list(package = ``BRACoD")))

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1.9001

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Adrian Verster [aut, cre]

Maintainer Adrian Verster <`adrian.verster@hc-sc.gc.ca`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2022-03-24 15:10:07 UTC

Contents

`convergence_tests` *Perform convergence tests on the p and beta variables*

Description

You may get errors are divergence of some variables after pymc3 samples the posterior. We are not overly concerned about some of the variables, such as the variance, rather we are really interested in the inclusion probabilities (p) and contribution coefficients (beta). The convergence tests that are included here focus on evaluating those two variables.

Usage

```
convergence_tests(trace, df_relab)
```

Arguments

`trace`	the output of run_bracod()
`df_relab`	the microbiome relative abundance

Value

no return value

`install_bracod` *Install BRACoD in python*

Description

Uses pip to install the latest BRACoD release in python. You might need to specify a python environment with either reticulate::use_virtualenv or reticulate::use_condaenv.

Usage

```
install_bracod(method = "auto", conda = "auto")
```

Arguments

`method`	passed to reticulate::py_install
`conda`	passed to reticulate::py_install

Value

no return value

`obesity` *Example microbiome data*

Description

This data is mouse stool microbiome data from a study of obesity.

Usage

```
data(obesity)

df_scfa
```

Format

a DataFrame of 16S microbiome counts, and a dataframe with corresponding butyrate measurements

An object of class `data.frame` with 119 rows and 1 columns.

`remove_null` *Remove NULL values in your OTU and environmental variable*

Description

This will remove samples that are NULL in the environmental variable, as well as the corresponding samples in your relative abundance data.

Usage

```
remove_null(df_relab, Y)
```

Arguments

`df_relab`	microbiome relative abundance data in a dataframe
`Y`	values of the environmental variable

Value

a list containing 1) the relative abundance data and 2) the Y values

`run_bracod` *Run the main BRACoD algorithm*

Description

Uses pymc3 to sample the posterior of the model to determine bacteria that are associated with your environmental variable.

Usage

```
run_bracod(df_relab, env_var, n_sample = 1000, n_burn = 1000, njobs = 4)
```

Arguments

`df_relab`	A dataframe of relative microbiome abundances. Samples are rows and bacteria are columns.
`env_var`	the environmental variable you are evaluating. You need 1 measurement associated with each sample.
`n_sample`	number of posterior samples.
`n_burn`	number of burn-in steps before actual sampling stops.
`njobs`	number of parallel MCMC chains to run.

Value

the pymc trace object which holds the samples of the posterior distribution

Examples

```
## Not run:
data(obesity)
r <- simulate_microbiome_counts(obesity)
sim_counts <- r[[1]]
sim_y <- r[[2]]
contributions <- r[[3]]
sim_relab <- scale_counts(sim_counts)
trace <- run_bracod(sim_relab, sim_y, n_sample = 1000, n_burn=1000, njobs=4)

## End(Not run)
```

`scale_counts` *Normalize OTU counts and add a pseudo count*

Description

BRACoD requires relative abundance and cannot handle zeros, so this function adds a small pseudo count (1/10th the smallest non-zero value).

Usage

```
scale_counts(df_counts)
```

Arguments

`df_counts`	A dataframe of OTU counts. Samples are rows and bacteria are columns.

Value

a dataframe of relative abundance data

`score` *Score the results of BRACoD*

Description

This calculate the precision, recall and F1 of your BRACoD results if you know the ground truth, ie. if this is simulated data.

Usage

```
score(taxon_identified, taxon_actual)
```

Arguments

`taxon_identified`	a list of integers corresponding to the indicies of the taxon you identified with BRACoD
`taxon_actual`	a list of integers corresponding to the indicies of the taxon that truely contribute to butyrate levels

Value

a list containing 1) the precision 2) the recall 3) the f1 metric

Examples

```
## Not run:
df_summary <- summarize_trace(trace, colnames(sim_relab))
taxon_identified <- df_summary$taxon
taxon_actual <- which(contributions != 0)

r <- score(taxon_identified, taxon_actual)

precision <- r[[1]]
recall <- r[[2]]
f1 <- r[[3]]

print(sprintf("Precision: %.2f, Recall: %.2f, F1: %.2f",precision, recall, f1))

## End(Not run)
```

`simulate_microbiome_counts` *Simulate microbiome counts*

Description

Each bacteria's absolute abundance is simulated from a lognormal distribution. Then, convert each sample to relative abundance, and simulate sequencing counts using a multinomial distribution, based on the desired number of reads and the simulated relative abundances. This also simulates an environmental variable that is produced by some of the bacteria.

Usage

```
simulate_microbiome_counts(
  df,
  n_contributors = 20,
  coeff_contributor = 0,
  min_ab_contributor = -9,
  sd_Y = 1,
  n_reads = 1e+05,
  var_contributor = 5,
  use_uniform = TRUE,
  n_samples_use = NULL,
  corr_value = NULL,
  return_absolute = FALSE,
  seed = NULL
)
```

Arguments

`df`	A dataframe of OTU counts that is a model for data simulation. Samples are rows and bacteria are columns.
`n_contributors`	the number of bacteria that are to contribute to your environmental variable.
`coeff_contributor`	the average of the distribution used to simulate the contribution coefficient.
`min_ab_contributor`	The minimum log relative abundance, averaged across samples, to include a bacteria
`sd_Y`	the standard deviation of the simulated environmental variable
`n_reads`	the number of reads to be simulated per sample
`var_contributor`	If you use a uniform distribution, this is the range of the distribution, with a normal distribution it is the variance used to simulate the contribution coefficient.
`use_uniform`	use a uniform distribution to simulate the contribution coefficient. Alternative is the normal distribution.
`n_samples_use`	number of microbiome samples to simulate. If NULL, uses the same number of samples as in your dataframe
`corr_value`	the bacteria-bacteria correlation value you want to include in the simulation
`return_absolute`	returns the abosulte abundance values instead of the simulated microbiome counts
`seed`	random seed for reproducibility

Value

a list containing 1) the simulated count data 2) the simulated environmental variable and 3) the simulated contribution coefficients

---

`summarize_trace` *Summarize the results of BRACoD*

Description

This summarizes the trace object that run_bracod() returns. It returns a dataframe that contains two parameters of interest, the average inclusion (p) and the average coefficient (beta), telling you the association between that bacteria and the environmental variable

Usage

```
summarize_trace(trace, taxon_names = NULL, cutoff = 0.3)
```

Arguments

`trace`	the pymc3 object that is the output of run_bracod()
`taxon_names`	optional, a list of names of the bacteria to include in the results
`cutoff`	this is the cutoff on the average inclusion for inclusion. We reccomend a value of 0.3, but you can lower the value to include less confident taxon or raise the cutoff to exclude them.

Value

a dataframe with information about the bacteria that BRACoD identified

Examples

```
## Not run:
trace <- run_bracod(sim_relab, sim_y, n_sample = 1000, n_burn=1000, njobs=4)
df_summary <- summarize_trace(trace, colnames(sim_relab))

## End(Not run)
```

`threshold_count_data` *Threshold your microbiome counts data*

Description

This function removes samples below a minimum counts and bacteria below a minimum log abundance. Run this before running BRACoD because the algorithm does not perform well when there are many low abundance bacteria that are only present in a few samples.

Usage

```
threshold_count_data(df_counts, min_counts = 1000, min_ab = 1e-04)
```

Arguments

`df_counts`	A dataframe of OTU counts. Samples are rows and bacteria are columns.
`min_counts`	threshold samples with fewer than this many counts
`min_ab`	threshold bacteria whose average log abundance is below this

Value

a dataframe of microbiome counts

Index