Package 'themetagenomics'

June 6, 2017

Title Exploring Thematic Structure and Predicted Functionality of 16s rRNA Amplicon Data

Version 0.1.0

Description A means to explore the structure of 16S rRNA surveys using a Structural Topic Model coupled with functional prediction. The user provides an abundance table, sample metadata, and taxonomy information, and themetagenomics infers associations between topics and sample features, as well as topics and predicted functional content. Functional prediction can be accomplished via Tax4Fun (for Silva references) and PICRUSt (for GreenGeenes references).

URL http://github.com/EESI/themetagenomics

BugReports http://github.com/EESI/themetagenomics/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.2.5), Rcpp (>= 0.11.3)

Imports ggplot2, lda, lme4 (>= 1.1.12), Matrix, plotly (>= 4.5.6), rstan (>= 2.14.0), scales, shiny (>= 1.0.0), stats, stats4, stm (>= 1.1.4),

Suggests assertthat, covr, huge, igraph, inline, knitr, networkD3, proxy, rmarkdown, RcppArmadillo, Rtsne, testthat, vegan, viridis,

LinkingTo Rcpp

RoxygenNote 6.0.1.9000

VignetteBuilder knitr

NeedsCompilation yes

Author Stephen Woloszynek [aut, cre]

Maintainer Stephen Woloszynek <sw424@drexel.edu>

Repository CRAN

Date/Publication 2017-06-06 19:00:07 UTC

R topics documented:

---

cnn *Normalize an OTU table by 16S rRNA copy number*

Description

Implements 16S rRNA copy number normalization using the PICRUSt 16S GreenGreenes 13.5 copy number count table (default) or a user provided set of copy numbers.

Usage

```
cnn(otu_table, rows_are_taxa, copy_numbers, drop = TRUE, verbose = FALSE)
```

Arguments

otu_table	(required) Matrix or dataframe containing taxa abundances (counts, non-negative integers) across samples. Rows and columns must be uniquely named.
rows_are_taxa	(required) Logical flag indicating whether otu_table rows correspond to taxa (TRUE) or samples (FALSE).
copy_numbers	A 2-column matrix or data frame of copy numbers where column 1 contains the OTU IDs and column 2 the copy numbers.
drop	Logical flag to drop empty rows and columns. Defaults to TRUE.
verbose	Logical flag to print progress information. Defaults to FALSE.

Value

A normalized, rounded (to nearest integer) abundance table.

References

Langille, M. G.I.*, Zaneveld, J.*, Caporaso, J. G., McDonald, D., Knights, D., a Reyes, J., Clemente, J. C., Burkepile, D. E., Vega Thurber, R. L., Knight, R., Beiko, R. G., and Huttenhower, C. (2013). Nature Biotechnology, 1-10. 8.

Examples

```
nOTU <- cnn(GEVERS$OTU,rows_are_taxa=FALSE,drop=TRUE)
```

DAVID *Human logitudinal microbiome data*

Description

This includes 16S amplicon sequencing measurements over time from 2 individuals. One donor provided both gut and oral samples, whereas the other donor provided only gut samples. The abundance table was generated via Dada2 using the Silva reference database. The data span 350 time points.

Usage

```
DAVID
```

Format

A list containing a 746x1493 matrix (ABUND), a 1493x7 matrix (TAX), and a 746x9 dataframe (META).

Source

BioProject: PRJEB6518 (PubMed)

References

David, L. A., Materna, A. C., Friedman, J., Campos-Baptista, M. I., Blackburn, M. C., Perrotta, A., Erdman, S. E., and Alm, E. J. (2014). Genome Biology. 15:R89.

Examples

```
hist(log(DAVID$ABUND + 1),100)
table(DAVID$META$Site,DAVID$META$Donor)
```

download_ref *Download functional prediction reference tables*

Description

A function to download the KO and COG 13.5 GreenGenes reference tables for PICRUSt prediction or the KO reference table for tax4fun prediction. The data are stored at https://gitlab.com/sw1/themetagenomics_data/.

Usage

```
download_ref(destination, reference = c("all", "gg_ko", "gg_cog", "silva_ko"),
  overwrite = FALSE, verbose = FALSE)
```

Arguments

destination	Location of the folder to save the reference files.
reference	A string for either gg_ko, gg_cog, silva_ko, or all. Defaults to all.
overwrite	Logical flag to overwrite if file already exists. Default to FALSE.
verbose	Logical flag to print progress information. Defaults to FALSE.

See Also

picrust t4f

Examples

```
## Not run:
download_ref(destination='/references',reference='gg_ko')

## End(Not run)
```

est *Estimate topic or function effects*

Description

Estimate topic or function effects

Usage

```
est(object, ...)
```

Arguments

object	A topics or functions object generated by find_topics or predict, respectively.
...	Additional arguments for methods.

See Also

est.topics est.functions est.hmc est.ml

---

est.functions *Estimate predicted function-topic effects*

Description

Given within topic functional predictions, estimate the effects at a given gene function category level. The effects correspond to a topic-gene category interaction term after accounting for topic and gene category effects. The model can be fit via either maximum likelihood or Hamiltonian MC.

Usage

```
## S3 method for class 'functions'
est(object, topics_subset, level = 2, method = c("hmc",
  "ml"), seed = object$seeds$next_seed, verbose = FALSE, ...)

## S3 method for class 'hmc'
est(object, inits, prior = c("t", "normal", "laplace"),
  t_df = c(7, 7, 7), iters = 300, warmup = iters/2, chains = 1,
  cores = 1, seed = sample.int(.Machine$integer.max, 1),
  return_summary = TRUE, verbose = FALSE, ...)

## S3 method for class 'ml'
est(object, iters = 1000, verbose = FALSE,
  seed = sample.int(.Machine$integer.max, 1), ...)
```

Arguments

object	(required) Ouput of predict.topics.
topics_subset	Vector of topic indexes to be evaluated. Recommended to be < 25.
level	Gene category level to evalulate. Defaults to 2.
method	String indicating either ml or hmc. Defaults to hmc.
seed	Seed for the random number generator to reproduce previous results.
verbose	Logical flag to print progress information. Defaults to FALSE.
...	Additional arguments for methods.
inits	List of values for parameter initialization. If omitted, values are generated via glmer.nb

`prior`	Prior to be placed on covariate weights. Choices include student-t, normal, and laplace. Defaults to student-t.
`t_df`	Degrees of freedom for student-t priors. Defaults to 7.
`iters`	Number of iterations for for fitting. Defaults to 300 and 100 for HMC and ML, respectively.
`warmup`	For HMC, proportion of iterations devoted to warmup. Defaults to iters/2.
`chains`	For HMC, number of independent chains. Defaults to 1.
`cores`	For HMC, number of cores to parallelize chains. Defaults to 1.
`return_summary`	Logical flag to return results summary. Defaults to TRUE.

Details

The functional effects are estimated via a multilevel Bayesian negative binomial regression model. Topic and pathway level effects are estimated, as well as topic-pathway interactions. The model has the following form:

$$\theta_i = \mu + \beta_w + \beta_k + \beta_{w,k}$$

$$y_i \; NB(\theta_i, \phi)$$

where μ is the intercept and each β term represents the weight for pathway level, topic, and pathway level-topic interaction, respectively; ϕ represents the dispersion parameter.

HMC: Hamiltonian MC is performed via Stan. By default, student-t priors with degrees of freedom set at 7 are placed on all regression weights, with variance terms distributed by half normal priors. The intercept μ is given a normal prior with fixed variance. Lastly, ϕ is given an $exponential(.5)$ prior. The priors placed on the regression weights can be changed by the user to either normal, t-family, or laplace (double exponential) priors if a sparse solution is desired. For the latter, each variance term is given an additional regularization parameter λ which in turn is distributed by a $chi - squared(1)$ distribution.

Unless a set of initization values are provided by the user, or the user chooses to select a random initialization procedure, initial values are set at the maximum likelihood estimate via `glmer.nb`, but at a far smaller number of iterations than had the user chosen ML as his or her estimation method.

ML: Maximum likelihood estimation is performed via `glmer.nb`. For deeper level functional categories, the model may fail to converge, even with a substantial number of iterations. In such a case, the model estimates are returns so the user can perform HMC, but by initializing at these ML values.

Value

An object of class effects containing

model List containing the parameters, fit, and summary.

gene_table Dataframe containing the formatted predicted gene information from `predict.topics`.

References

Bates, D., Maechler, M., Bolker, B., and Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.

Gelman, A. and Hill, J. (2006). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press; 1 edition.

Stan Development Team. 2016. RStan: the R interface to Stan. http://mc-stan.org

Stan Development Team. 2016. Stan Modeling Language Users Guide and Reference Manual, Version 2.14.0. http://mc-stan.org

See Also

glmer.nb stan resume

Examples

```
formula <- ~DIAGNOSIS
refs <- 'CD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)

functions <- predict(topics,reference_path='/references/ko_13_5_precalculated.tab.gz')
function_effects <- est(functions,level=3,
                        iters=500,method='hmc',
                        prior=c('laplace','t','laplace'))

## End(Not run)
```

---

est.topics *Estimate topic effects*

Description

Given a covariate of interest, measure its relationship with the samples over topics distribution from the STM.

Usage

```
## S3 method for class 'topics'
est(object, metadata, formula, refs, nsims = 100,
  ui_level = 0.8, npoints = 100, seed = object$seeds$next_seed,
  verbose = FALSE, ...)
```

Arguments

object	(required) Ouput of find_topics.
metadata	Matrix or dataframe containing sample information with row or column names corresponding to the otu_table.
formula	New formula for covariates of interest found in metadata, different than the formula used to generate object. Interactions, transformations, splines, and polynomial expansions are permitted.
refs	Character vector of length equal to the number of factors or binary covariates in formula, indicating the reference level.
nsims	Number of simulations to perform for estimating covariate effects. Defaults to 100.
ui_level	Width of uncertainty interval for reporting effects. Defaults to .95.
npoints	Number of posterior predictive samples to draw. Defaults to 100.
seed	Seed for the random number generator to reproduce previous results.
verbose	Logical flag to print progress information. Defaults to FALSE.
...	Additional arguments for methods.

Details

The posterior predictive estimates are calculated depending on the type of covariate. First, all factors are expanded using dummy variables, setting the reference classes as intercepts. For each topic, the topic frequency over samples is regressed against the expanded design matrix. Covariate weights and the variance-covariance matrix is then calculated, which are used to sample new weights using a multivariate normal distribution.

The estimation of a specific covariate effect is performed by calculated y-hat from the posterior predictive distribution by holding all covariates other than the target covariate fixed. This is accomplished by marginalizing over the sample data. This fixed design matrix is then multiplied by the weights simulated from the multivariate normal distribution. For a target binary covariate x (which includes expanded factors), effect estimates are defined as the difference between y-hat when x=1 and y-hat when x=0 is calcuated, with the reference covariate designated as 1 (hence negative differences imply a strong effect for the reference class). For continuous covariates, the effect estimates are defined as the regression weight for that covariate of interest. To explore the posterior predictive distribution, y-hat is again calculated, but over a vector of values spanning the range of the continuous covariate, with other covariates held fixed as before. Additional y-hat are then calculated while iteratively setting each binary covariate to 0, to explore their influence on the continuous covariate. Nonlinear covariates (e.g., splines) are treated similarly with respect to y-hat. Their effect estimates, however, are calculated by calculating the Spearman rank correlation coefficient between y-hat and y.

For each covariate, the effect estimate is returned. y-hat vectors are returned as well for continuous and nonlinear covariates. All effect estimates are ranked in terms of weight or correlation coefficient. Values not overlapping 0 given a user designed level of uncertainty or returned as "significant."

Value

An object of class effects containing

topic_effects List of the effect estimates for the covariates in formula.

topics Object of class topics containing the original output of find_topics.

modelframe Original modelframe.

References

Gelman, A. and Hill, J. (2006). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press; 1 edition.

Roberts, M.E., Stewart, B.M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S.K., Albertson, B., & Rand, D.G. (2014). Structural topic models for open-ended survey responses. Am. J. Pol. Sci. 58, 1064–1082.

See Also

estimateEffect

Examples

```
formula <- ~DIAGNOSIS
refs <- 'CD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)
topic_effects <- est(topics)

## End(Not run)
```

---

extract *Extract summary statistics*

Description

Extract summary statistics

Usage

```
extract(object, ...)

## S3 method for class 'effects'
extract(object, ...)
```

Arguments

`object`	Object of class effects, fit via hmc.
`...`	Additional arguments for methods.

Value

A list containing

summary Rstan summary of parameters from model.

flagged Vector of parameter names with Rhat > 1.1.

Methods (by class)

- `effects`: Extract summary statistics from HMC effects object
  Extracts the summary information in a form condusive with vis methods, specifically in cases when return_summary was set to FALSE.

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)

functions <- predict(topics,reference_path='/references/ko_13_5_precalculated.tab.gz')

function_effects <- est(functions,level=3,
                        iters=500,method='hmc',
                        prior=c('laplace','t','laplace'),
                        return_summary=FALSE)

function_effects_summary <- extract(function_effects)

## End(Not run)
```

`find_topics` *Perform topic estimation on a themetadata object*

Description

Given a themetadata object, this function converts the OTU counts across samples into a document format and then fits a structural topic model by wrapping the stm function from pacakge stm.

Usage

```
find_topics(themetadata_object, K, sigma_prior = 0, model = NULL,
  iters = 500, tol = 1e-05, batches = 1, init_type = c("Spectral",
  "LDA", "Random"), seed = themetadata_object$seed, verbose = FALSE,
  verbose_n = 5, control = list())
```

Arguments

themetadata_object	(required) Ouput of prepare_data.
K	(required) A positive integer indicating the number of topics to be estimated.
sigma_prior	Scalar between 0 and 1. This sets the strength of regularization towards a diagonalized covariance matrix. Setting the value above 0 can be useful if topics are becoming too highly correlated. Defaults to 0.
model	Prefit STM model object to restart an existing model.
iters	Maximum number of EM iterations. Defaults to 500.
tol	Convergence tolerance. Defaults to 1e-5.
batches	Number of groups for memorized inference. Defaults to 1.
init_type	Type of initialization procedure. Defaults to Spectral
seed	Seed for the random number generator to reproduce previous results.
verbose	Logical flag to print progress information. Defaults to FALSE.
verbose_n	Integer determining the intervals at which labels are printed.
control	List of additional parameters control portions of the optimization. See details.

Details

Topics are estimated via stm from the stm package. The focus of the themetagenomics pipeline is leveraging both abundance and predicted functional information of 16S rRNA sequencing; hence, the pipeline calls for the use of only "prevalence" information (to use stm terminology). This wrapper therefore removes any options pertaining to "content." If the user is interested in exploring the content component of the STM, then the stm package itself is the ideal place to start. Given that only the prevalence component can be manipulated using find_topics, the following additional parameters can be passed to control as a list (adapted from stm documentation):

gamma.enet Scalara between 0 and 1 that controls the degree of L1 and L2 regularization, where 0 and 1 correspond to ridge and lasso regression. Defaults to 1.

gamma.ic.k Method to select the regularization parameter where 2 corresponds to AIC and log(n) is equivalent to BIC. Defaults to 2.

gamma.maxits Maximum number of iterations for estimating prevalence. Defaults to 1000.

nits For LDA initialization, the number of Gibbs sampling iterations. Defaults to 50.

burnin For LDA initialization, the number of burnin iterations. Defaults to 25.

alpha For LDA initialization, the samples over topics distribution hyperparameter.

eta For LDA initialization, the topics over words distribution hyperparameter.

rp.s For spectral initialization, scalar between 0 and 1 that controls the degree sparsity of random projections. Defaults to .05

rp.p For spectral initialization, the dimensinoality of random projections. Defaults to 3000.

rp.d.group.size For spectral initialization, the block size. Defaults to 2000.

maxV For spectral initialization, the maximum number of words used during initialization.

Value

An object of class topics containing

fit STM object containing topic model fit

docs Abundance table in document form of length equal to the number of samples. Each element contains 2-row array, where row 1 contains the the vocabulary index of a given taxon and row 2 contains its abundance in that document

vocab Character vector containing vocabulary of taxa IDs, where their position corresponds to the document indexes

otu_table Original otu_table

tax_table Original tax_table

metadata Original metadata

ref Original covariate references

modelframe Original modelframe

splineinfo Original splineinfo

References

Roberts, M.E., Stewart, B.M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S.K., Albertson, B., & Rand, D.G. (2014). Structural topic models for open-ended survey responses. Am. J. Pol. Sci. 58, 1064–1082.

See Also

glmnet stm

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)

## End(Not run)
```

GEVERS *Inflammatory bowel disease gut microbiome data*

Description

Subset of samples from the 16S amplicon Gevers et al prediatric inflammatory bowel disease (IBD) dataset. The data include 200 gut samples, 100 of which are controls, spanning 991 OTUs. Three tables are included: an OTU table generated via QIIME, picked against GreenGreens 13.5 at 97 similarity; a taxonomy reference table, and a sample metadata table that includes diagnoiss and PCDAI scores, a continuous measure of disease burden.

Usage

```
GEVERS
```

Format

A list containing a 200x991 matrix (OTU), a 991x7 matrix (TAX), and a 200x3 dataframe (META).

Source

BioProject: PRJNA237362. (PubMed)

References

Gevers, D., Kugathasan, S., Denson, L.A., et al. (2014). The Treatment-Naive Microbiome in New-Onset Crohn's Disease. Cell Host Microbe 15, 382–392. (PubMed)

Examples

```
hist(log(GEVERS$OTU + 1),100)
table(GEVERS$META$DIAGNOSIS)
boxplot(subset(GEVERS$META,DIAGNOSIS == 'CD')[['PCDAI']])
```

picrust *Predict OTU functional content via PICRUSt*

Description

Given an OTU abundance table prepared with the GreenGenes reference database, this function predicts the functional content using either COG or KO precalculated mapping tables that map the taxonomic abundance for a given OTU to functional abundance content across a set of functional genes.

Usage

```
picrust(otu_table, rows_are_taxa, reference = c("gg_ko", "gg_cog"),
  reference_path, cn_normalize = FALSE, sample_normalize = FALSE,
  drop = TRUE)
```

Arguments

otu_table	(required) Matrix or dataframe containing taxa abundances (counts, non-negative integers) across samples. Rows and columns must be uniquely named.
rows_are_taxa	(required) Logical flag indicating whether otu_table rows correspond to taxa (TRUE) or samples (FALSE).
reference	A string for either gg_ko or gg_cog. Defaults to gg_ko.
reference_path	Folder path of the reference file
cn_normalize	Logical flag for performing 16S rRNA copy number normalization. Defaults to FALSE.
sample_normalize	Logical flag to normalize functional predictions by the total functional abundance in a sample. Defaults to FALSE.
drop	Logical flag to drop empty gene columns. Defaults to TRUE.

Value

A list containing

fxn_table A matrix of gene counts across topics.

fxn_meta A list of functional metadata corresponding to fxn_table.

method_meta A matrix of method specific metadata (NSTI).

References

Langille, M. G.I.*, Zaneveld, J.*, Caporaso, J. G., McDonald, D., Knights, D., a Reyes, J., Clemente, J. C., Burkepile, D. E., Vega Thurber, R. L., Knight, R., Beiko, R. G., and Huttenhower, C. (2013). Nature Biotechnology, 1-10. 8.

See Also

download_ref picrust

Examples

```
## Not run:
download_ref(destination='/references',reference='gg_ko')

predicted_functions <- picrust(otu_table=GEVERS$OTU,rows_are_taxa=TRUE,
                               reference='gg_ko',reference_path='/references',
                               cn_normalize=TRUE,
                               sample_normalize=FALSE,drop=TRUE)
```

```
## End(Not run)
```

`picrust_otu` *Predict OTU functional potential via PICRUSt*

Description

Given an OTU abundance table prepared with the GreenGenes reference database, this function predicts the functional content using either COG or KO precalculated mapping tables that map the taxonomic abundance for a given OTU to functional abundance content across a set of functional genes.

Usage

```
picrust_otu(file_path, otu_id_targets)
```

Arguments

`file_path`	Path to the precalculated table
`otu_id_targets`	Character vector of OTU IDs to predict

Value

A list containing

gene_ids String vector of KO IDs, the column names in genome_table_out.

pimeta_ids String vector of names for the PICRUSt metdata categories, the column names of pimeta_table_out.

matches String vector of OTU IDs from otu_id_targets that were present in the mapping file.

genemeta String vector of functional metadata corresponding to gene_ids

genome_table_out Integer matrix of gene counts across topics

pimeta_table_out Numeric matrix of method specific metadata (NSTI)

`predict.topics` *Predict topic functional content*

Description

Given an object of class topics, this function predicts the functional content using PICRUSt or tax4fun precalculated mapping tables that maps the taxonomic abundance for a given OTU to functional abundance content across a set of functional genes.

Usage

```
## S3 method for class 'topics'
predict(object, reference = c("gg_ko", "gg_cog", "silva_ko"),
  reference_path, scalar = 100, cn_normalize = FALSE,
  sample_normalize = FALSE, drop = TRUE, ...)
```

Arguments

`object`	(required) Output of `find_topics`.
`reference`	A string for either gg_ko, gg_cog, or silva_ko. Defaults to gg_ko.
`reference_path`	Folder path of the reference file
`scalar`	Value for scaling the topics over taxa distrubution to predicted counts. Defaults to 100.
`cn_normalize`	Logical flag for performing 16S rRNA copy number normalization. Defaults to FALSE.
`sample_normalize`	Logical flag to normalize functional predictions by the total functional abundance in a sample. Defaults to FALSE.
`drop`	Logical flag to drop empty gene columns. Defaults to TRUE.
`...`	Additional arguments for t4f method.

Value

An object of class functions containing

fxn_table A matrix of gene counts across topics.

fxn_meta A list of functional metadata corresponding to fxn_table.

method_meta A matrix of method specific metadata.

References

ABhauer, K. P., Wemheuer, B. Daniel, R., and Meinicke, P. (2015). Bioinformatics, 1-3. 31(17).

Langille, M. G.I.*, Zaneveld, J.*, Caporaso, J. G., McDonald, D., Knights, D., a Reyes, J., Clemente, J. C., Burkepile, D. E., Vega Thurber, R. L., Knight, R., Beiko, R. G., and Huttenhower, C. (2013). Nature Biotechnology, 1-10. 8.

See Also

download_ref picrust t4f

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)

download_ref(destination='/references',reference='gg_ko')
functions <- predict(topics,reference='gg_ko',
                     reference_path='/references')

## End(Not run)
```

---

prepare_data *Prepare themetadata object from data for topic modeling pipeline*

Description

Creates a themetadata class by preprocessing data from an OTU table, taxonimic information, sample metadata, and a formula reflecting the preposed relationship between sample metadata and the topics over samples distribution.

Usage

```
prepare_data(otu_table, rows_are_taxa, tax_table, metadata, formula, refs,
  cn_normalize = TRUE, drop = TRUE,
  seed = sample.int(.Machine$integer.max, 1), verbose = FALSE)
```

Arguments

otu_table	(required) Matrix or dataframe containing taxa abundances (counts, non-negative integers) across samples. Rows and columns must be uniquely named.
rows_are_taxa	(required) Logical flag indicating whether otu_table rows correspond to taxa (TRUE) or samples (FALSE).
tax_table	Matrix or dataframe containing taxonimc information with row or column names corresponding to the otu_table.
metadata	Matrix or dataframe containing sample information with row or column names corresponding to the otu_table.

formula	Formula for covariates of interest found in metadata. Interactions, transformations, splines, and polynomial expansions are permitted.
refs	Character vector of length equal to the number of factors or binary covariates in formula, indicating the reference level.
cn_normalize	Logical flag for performing 16S rRNA copy number normalization. Defaults to TRUE.
drop	Logical flag to drop empty rows and columns. Defaults to TRUE.
seed	Seed for random number generation. This seed will be passed to each function that uses this prepared data unless otherwise overridden. Defaults to a random integer between 1 and the maximum integer supported by R.
verbose	Logical flag to print progress information. Defaults to FALSE.

Value

An object of class themetadata containing

otu_table Matrix of taxa abundances, correctly overlapping with tax_table and metadata. Will be copy number normalized, lacking empty rows and columns by default.

tax_table Matrix, correctly overlapping with otu_table

metadata Dataframe, correctly overlapping with otu_table and formula. All character covariates are converted to factors.

formula Unaltered, given by the user

splineinfo List containing the covariate, nonlinear function name, and basis function expansion of all applicable covariates based on the formula.

modelframe Dataframe of metadata of only applicable covariates with factors expanded as dummy variables

See Also

s

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)
```

resume *Resume HMC using a previous fit*

Description

Perform HMC using a previously compiled Stan model. This is specifcally useful in cases when a previous fit failed to converged (i.e., Rhat > 1.1 for a portion of parameter estimates), thus requiring more iterations.

Usage

```
resume(object, ...)

## S3 method for class 'effects'
resume(object, init_type = c("last", "orig"), inits, iters,
  warmup = iters/2, chains = 1, cores = 1,
  seed = object$seeds$next_seed, return_summary = TRUE, verbose = FALSE,
  ...)
```

Arguments

object	(required) Ouput of est.functions.
...	Additional arguments for methods.
init_type	Type of initial parameters, either the original set that was passed to est.functions or the last parameter sample from the reused fit. Defaults to last.
inits	List of values for parameter initialization. Overrides init_type.
iters	Number of iterations for for fitting. Defaults to 300 and 100 for HMC and ML, respectively.
warmup	For HMC, proportion of iterations devoted to warmup. Defaults to iters/2.
chains	For HMC, number of parallel chains. Defaults to 1.
cores	For HMC, number of cores to parallelize chains. Defaults to 1.
seed	Seed for the random number generator to reproduce previous results.
return_summary	Logical flag to return results summary. Defaults to TRUE.
verbose	Logical flag to print progress information. Defaults to FALSE.

Value

An object of class effects containing

model List containing the parameters, fit, and summary.

gene_table Dataframe containing the formatted predicted gene information from predict.topics.

References

Stan Development Team. 2016. RStan: the R interface to Stan. http://mc-stan.org

See Also

stan est.functions

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
topics <- find_topics(dat,K=15)

functions <- predict(topics,reference_path='/references/ko_13_5_precalculated.tab.gz')

function_effects_init <- est(functions,level=3,iters=150,
                             prior=c('laplace','t','laplace'))
function_effects <- resume(function_effects_init,init_type='last',
                           iters=300,chains=4)

## End(Not run)
```

---

s *Make a B-spline Basis Function (from s)*

Description

This is a simple wrapper around the bs function in the splines package. It will default to a spline with 10 degrees of freedom.

Usage

```
s(x, df, ...)
```

Arguments

x	The predictor value.
df	Degrees of freedom. Defaults to the minimum of 10 or one minus the number of unique values in x.
...	Arguments passed to the bs function.

Details

This is a simple wrapper written as users may find it easier to simply type s rather than selecting parameters for a spline. We also include predict and makepredictcall generic functions for the class so it will work in settings where predict is called.

Value

A predictor matrix of the basis functions.

See Also

bs ns

---

t4f *Predict taxonomic functional content via tax4fun*

Description

Given a taxonomic abundance table prepared with the Silva reference database, predicts the functional content using a KO precalculated mapping table that maps the taxonomic abundance for a given tax_table to functional abundance content across a set of functional genes.

Usage

```
t4f(otu_table, rows_are_taxa, tax_table, reference_path, type = c("uproc",
  "pauda"), short = TRUE, cn_normalize = FALSE, sample_normalize = FALSE,
  scalar, drop = TRUE, verbose = FALSE)
```

Arguments

otu_table	(required) Matrix or dataframe containing taxa abundances (counts, non-negative integers) across samples. Rows and columns must be uniquely named.
rows_are_taxa	(required) Logical flag indicating whether otu_table rows correspond to taxa (TRUE) or samples (FALSE).
tax_table	Matrix or dataframe containing Silva taxonimic information with row or column names corresponding to the otu_table. Silva species information is required.
reference_path	Folder path of the silva-to-kegg mapping file (t4f_silva_to_kegg.rds) and reference profiles (t4f_ref_profiles.rds). Must not be renamed.
type	Type of protein domain classifcation methods used to generate references (uproc or pauda). Defaults to uproc.
short	Logical flag whether to use a short or long read references. Defaults to TRUE.
cn_normalize	Logical flag for performing 16S rRNA copy number normalization. Defaults to FALSE.
sample_normalize	Logical flag to normalize functional predictions by the total functional abundance in a sample. Defaults to FALSE.
scalar	Value for scaling the topics over functions distrubution to predicted counts.
drop	Logical flag to drop empty gene columns after prediction. Defaults to TRUE.
verbose	Logical flag to print progress information. Defaults to FALSE.

Value

A list containing

fxn_table A matrix of gene counts across topics.

fxn_meta A list of functional metadata corresponding to fxn_table.

method_meta A matrix of method specific metadata (FTU).

References

ABhauer, K. P., Wemheuer, B. Daniel, R., and Meinicke, P. (2015). Bioinformatics, 1-3. 31(17).

See Also

download_ref picrust

Examples

```
## Not run:
download_ref(destination='/references',reference='silva_ko')
predicted_functions <- t4f(otu_table=DAVID$OTU,rows_are_taxa=FALSE,
                           tax_table=DAVID$TAX,reference='/references',
                           type='uproc',short=TRUE,cn_normalize=TRUE,
                           sample_normalize=FALSE,scalar=NULL,drop=TRUE)

## End(Not run)
```

---

vis *Launch in interactive visualize to explore topic effects*

Description

Launch in interactive visualize to explore topic effects

Usage

```
vis(object, ...)

## S3 method for class 'effects'
vis(object, topic_effects, type = c("taxa", "binary",
  "continuous", "functions"), seed = object$seed$next_seed, ...)

## S3 method for class 'binary'
vis(object, taxa_grp_n = 7, ...)

## S3 method for class 'continuous'
vis(object, lambda_step = 0.1, taxa_reg_n = 8, ...)
```

```
## S3 method for class 'functions'
vis(object, topic_effects, beta_min = 1e-05,
  ui_level = 0.8, gene_min = 0, pw_min = 20, ...)

## S3 method for class 'taxa'
vis(object, taxa_bar_n = 30, top_n = 7, method = c("huge",
  "simple"), corr_thresh = 0.01, lambda_step = 0.01, ...)

## S3 method for class 'topics'
vis(object, taxa_bar_n = 30, top_n = 7,
  method = c("huge", "simple"), corr_thresh = 0.01, lambda_step = 0.01,
  ...)
```

Arguments

object	(required) Output of find_topics, or est.topics, or est.functions.
...	Additional arguments for methods.
topic_effects	Output of est.topics.
type	Type of visualization to perform.
seed	Seed for the random number generator to reproduce previous results.
taxa_grp_n	Number of taxa group names to display (remaining are renamed to other). Defaults to 7.
lambda_step	Value designating the lambda stepsize for calculating taxa relevance. Recommended to be between .01 and .1. Defaults to .1.
taxa_reg_n	Number of most relevant taxa within topic to regress. Defaults to 8.
beta_min	Minimum probability in topics over taxa distribution to set to 0. Defaults to 1e-5.
ui_level	Uncertainty level for plot intervals. Defaults to .8.
gene_min	Mininum count for gene set table. Defaults to 0.
pw_min	Maximium number of pathways to show in heatmap. for Defaults to 20.
taxa_bar_n	Number of taxa to show in the frequency bar plot. Defaults to 30.
top_n	Number of taxonic groups to colorize in the frequency bar plot. Defaults to 7.
method	Method for estimating topic correlations links. Defaults to huge.
corr_thresh	Threshold to set correlations to 0 when method is set to simple. Defaults to .01.

Details

Taxa: Integrates the samples over topics p(s|k) and topics over taxa p(k|t) distributions from the STM, the topic correlations from the p(s|k) component, the covariate effects from the p(s|k) component, and their relationship with the raw taxonomic abundances. The covariate effects for each topic are shown as a scatterplot of posterior weights with error bars corresponding the global approximation of uncertainty. If the covariate chosen is binary, then the posterior regression weights with uncertainty intervals are shown. This is analogous to the mean difference between factor

levels in the posterior predictive distribution. For continuous covariates, the points again represent the mean regression weights (i.e., the posterior slope estimate of the covariate). If, however, a spline or polynomial expansion was used, then the figure shows the posterior estimates of the standard deviation of the predicted topic probabilities from the posterior predictive distribution. Colors indicate whether a given point was positive (red) or negative (blue) and did not enclose 0 at a user defined uncertainty interval.

The ordination figure maintains the color coding just decribed. The ordination is performed on p(k|t) via either PCoA (using either Jensen-Shannon, Euclidean, Hellinger, Bray-Curtis, Jaccard, or Chi-squared distance) or t-SNE. The latter iterates through decreasing perplexity values (starting at 30) until the algorithm succeeds. The top 2 or 3 axes can be shown. The radius of the topic points corresponds to the topic frequencies marginalized over taxa.

The bar plot behaves in accordance with LDAvis. When no topics are chosen, the overall taxa frequencies are shown. These frequencies do not equal the abundances found in the initial abundance table. Instead, they show p(k|t) multiplied by the marginal topic distribution (in counts). To determine the initial order in which taxa are shown, these two distributions are compared via Kullback-Liebler divergence and then weighted by the overall taxa frequency. The coloration of the bars indiciates the taxonomic group the inidividual taxa belong to. The groups shown are determined based on the abundance of that group in the raw abundance table. When a topic is selected, the relative frequency of a given taxa in that topic is shown in red.

λ controls relevance of taxa within a topic, which in turn is used to adjust the order in which the taxa are shown when a topic is selected. Relevence is essentially a weighted sum between the probability of taxa in a given topic and the probability of taxa in a given topic relative to the overall frequency of that taxa. Adjusting λ influences the relative weighting such that

$$r = \lambda x log p(t|k) + \lambda x log p(t|k)/p(x)$$

The correlation graph shows the topic correlations from $p(s|k)\ MVN(mu, sigma)$. Again, the coloration described above is conserved. The size of the nodes reflects the magnitude of the covariate posterior regression weight, whereas the width of the edges represents the value of the positive correlation between the connected nodes. By default, the graph estimates are determined using the the huge package, which first performs a nonparanormal transformation of p(s|k), followed by a Meinhuasen and Buhlman procedure. Alternatively, by choosing the simple method, the correlations are simply a thresholded MAP estimate of p(s|k).

Binary: Integrates the topics over taxa p(k|t) distribution from the STM, binary covariate effects from the p(s|k) component, and their relationship with the raw taxonomic abundances. The covariate effects for each topic are shown as a scatterplot of posterior weights with error bars corresponding the global approximation of uncertainty. If the covariate chosen is binary, then the posterior regression weights with uncertainty intervals are shown. This is analogous to the mean difference between factor levels in the posterior predictive distribution. For continuous covariates, the points again represent the mean regression weights (i.e., the posterior slope estimate of the covariate). Colors indicate whether a given point was positive (red) or negative (blue) and did not enclose 0 at a user defined uncertainty interval.

Selecting a topic estimate generates violin plots showing the p(s|k) distribution, split based on chosen binary covariate effects. The slider allows the user to threshold the number of points shown, based on their values in p(s|k). Highlighting points in the violin plots generates bar plots that show their abundances (or relative abundanecs) in the raw abundance table.

Continuous: Integrates the samples over topics p(t|s) and the topics over taxa p(k|t) distributions from the STM, binary and continuous covariate effects from the p(s|k) component, and their re-

lationship with the raw taxonomic abundances. The covariate effects for each topic are shown as a scatterplot of posterior weights with error bars corresponding the global approximation of uncertainty. If the covariate chosen is binary, then the posterior regression weights with uncertainty intervals are shown. This is analogous to the mean difference between factor levels in the posterior predictive distribution. For continuous covariates, the points again represent the mean regression weights (i.e., the posterior slope estimate of the covariate). If, however, a spline or polynomial expansion was used, then the figure shows the posterior estimates of the standard deviation of the predicted topic probabilities from the posterior predictive distribution.

Selecting a topic estimate generates three panels. The top panel shows the posterior estimates of the selected continuous covariate. If binary covariates were present in the model formula, then the continuous effect given the binary covariate is shown as two regression lines, along with their corresponding uncertainty intervals. The points show the true p(k|s) values determined by the STM as a function of the selected continuous covariate. The middle panel then shows the raw abundances (or relative abundances) of most relavent taxa. Relavence can be control by adjusting λ where

$$r = \lambda x log p(t|k) + \lambda x log p(t|k)/p(x)$$

If binary covariates were provided in the model formula, selected split will split the regressions based on the selected covariate. Each figure overlays a linear best fit (red) and loess fit (red) to facilitate intepretation. The bottom panel shows these taxa combined.

Functions: Integrates the taxa over topics p(t|k) and gene functions over topics p(g|k) distributions, along with and the covariate effects from the p(s|k) component. The covariate effects for each topic are shown as a scatterplot of posterior weights with error bars corresponding the global approximation of uncertainty. IIf the covariate chosen is binary, then the posterior regression weights with uncertainty intervals are shown. This is analogous to the mean difference between factor levels in the posterior predictive distribution. For continuous covariates, the points again represent the mean regression weights (i.e., the posterior slope estimate of the covariate). If, however, a spline or polynomial expansion was used, then the figure shows the posterior estimates of the standard deviation of the predicted topic probabilities from the posterior predictive distribution. Colors indicate whether a given point was positive (red) or negative (blue) and did not enclose 0 at a user defined uncertainty interval.

The upper heatmap shows p(t|k), clustered via Wards method on a user chosen distance metric. Topics are ranked to right based on the weights from the aforementioned scatterplot. The lower heatmap shows the weights for the pathway-topic interaction from the multilevel Bayesian model. Positive and negative weight estimates that do not enclose zero at a chosen uncertainty level are marked with red and blue crosses, respectively. The pathway ordering is done via Wards method on Euclidean distance. Upon selected a cell within the pathway-topic heatmap, a table of genes is returned, ranking the genes in terms of abundance that belong to a given pathway-topic combination.

References

Roberts, M.E., Stewart, B.M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S.K., Albertson, B., & Rand, D.G. (2014). Structural topic models for open-ended survey responses. Am. J. Pol. Sci. 58, 1064–1082.

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. Proc. Work. Interact. Lang. Learn. Vis. Interfaces.

Zhao, T., & Liu., H. (2012) The huge Package for High-dimensional Undirected Graph Estimation in R. Journal of Machine Learning Research.

See Also

igraph_to_networkD3, huge, topicCorr, Rtsne

Examples

```
formula <- ~DIAGNOSIS
refs <- 'Not IBD'

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
vis(topic_effects,type='taxa')
vis(topic_effects,type='binary')

## End(Not run)

formula <- ~PCDAI

dat <- prepare_data(otu_table=GEVERS$OTU,rows_are_taxa=FALSE,tax_table=GEVERS$TAX,
                    metadata=GEVERS$META,formula=formula,refs=refs,
                    cn_normalize=TRUE,drop=TRUE)

## Not run:
vis(topic_effects,type='continuous')

functions <- predict(topics,reference_path='/references/ko_13_5_precalculated.tab.gz')

function_effects <- est(functions,level=3,
                        iters=500,method='hmc',
                        prior=c('laplace','t','laplace'))

vis(function_effects,topic_effects)

## End(Not run)
```

Index