# Package 'SIPmg'

February 22, 2023

**Title** Statistical Analysis to Identify Isotope Incorporating MAGs

**Version** 1.4.1

**Description** Statistical analysis as part of a stable isotope probing (SIP) metagenomics study to identify isotope incorporating taxa recovered as metagenome-assembled genomes (MAGs). Helpful reading and a vignette in bookdown format is provided on the package site <https://zielslab.github.io/SIPmg.github.io/>.

**URL** https://zielslab.github.io/SIPmg.github.io/

**BugReports** https://github.com/ZielsLab/SIPmg

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** HTSSIP, dplyr, lazyeval, phyloseq, plyr, stringr, tibble, tidyr, magrittr, ggplot2, ggpubr, purrr, rlang, MASS, DESeq2, data.table, utils

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Depends** R (>= 3.5.0)

**Suggests** rmarkdown, knitr, EBImage, readr, BiocManager

**LazyData** true

**Author** Pranav Sampara [aut, cre],
Kate Waring [ctb],
Ryan Ziels [aut]

**Maintainer** Pranav Sampara <pranav.sai.4@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-02-22 00:40:10 UTC

# R topics documented:

---

atomX                           *Atom fraction excess table*

---

### Description

Data table generated from the "qSIP_atom_excess_MAGs" function

### Usage

```
data(atomX)
```

### Format

An object of class "list"

---

calc_atom_excess_MAGs    *Calculate atom fraction excess*

---

### Description

See Hungate et al., 2015 for more details

### Usage

```
calc_atom_excess_MAGs(Mlab, Mlight, Mheavymax, isotope = "13C")
```

### Arguments

| | |
|---|---|
| Mlab | The molecular wight of labeled DNA |
| Mlight | The molecular wight of unlabeled DNA |
| Mheavymax | The theoretical maximum molecular weight of fully-labeled DNA |
| isotope | The isotope for which the DNA is labeled with ('13C' or '18O') |

### Value

numeric value: atom fraction excess (A)

---

calc_Mheavymax_MAGs    *Calculate Mheavymax*

---

### Description

This script was adapted from https://github.com/buckleylab/HTSSIP/blob/master/R/qSIP_atom_excess.R
for use with genome-centric metagenomics. See Hungate et al., 2015 for more details

### Usage

```
calc_Mheavymax_MAGs(Mlight, isotope = "13C", Gi = Gi)
```

### Arguments

| | |
|---|---|
| Mlight | The molecular wight of unlabeled DNA |
| isotope | The isotope for which the DNA is labeled with ('13C' or '18O') |
| Gi | The G+C content of unlabeled DNA |

### Value

numeric value: maximum molecular weight of fully-labeled DNA

---

coverage_normalization

> *Normalize feature coverages to estimate absolute abundance or relative coverage using MAG/contig coverage values with or without multiplying total DNA concentration of the fraction*

---

## Description

Normalize feature coverages to estimate absolute abundance or relative coverage using MAG/contig coverage values with or without multiplying total DNA concentration of the fraction

## Usage

```
coverage_normalization(
  f_tibble,
  contig_coverage,
  sequencing_yield,
  fractions_df,
  approach = "relative_coverage"
)
```

## Arguments

f_tibble
: Can be either of (1) a tibble with first column "Feature" that contains bin IDs, and the rest of the columns represent samples with bins' coverage values. (2) a tibble as outputted by the program "checkm coverage" from the tool CheckM. Please check CheckM documentation - https://github.com/Ecogenomics/CheckM on the usage for "checkm coverage" program

contig_coverage
: tibble with contig ID names ("Feature" column), sample columns with **same sample names as in f_tibble** containing coverage values of each contig, contig length in bp ("contig_length" column), and the MAG the contig is associated ("MAG" column) **with same MAGs as in Feature column of f_tibble dataset**.

sequencing_yield
: tibble containing sample ID ("sample" column) with **same sample names as in f_tibble** and number of reads in bp recovered in that sample ("yield" column).

fractions_df
: fractions data frame A fractions file with the following columns

- Replicate: Depends on how many replicates the study has
- Fractions: Typically in the range of 2-24
- Buoyant_density: As calculated from the refractometer for each fraction and replicate
- Isotope: "12C", "13C", "14N", "15N" etc.
- DNA_concentration
- Sample: In the format "'isotope'*rep*#*fraction*#". For instance, "12C_rep_1_fraction_1"

| | |
|---|---|
| approach | Please choose the method for coverage normalization as "relative_coverage", "greenlon", "starr" to estimate only relative coverage without multiplying DNA concentration of fraction, or as per methods in Greenlon et al. - https://journals.asm.org/doi/full/10.1128/n 22 or Starr et al. - https://journals.asm.org/doi/10.1128/mSphere.00085-21 |

## Value

tibble containing normalized coverage in required format with MAG name as first column and the normalized coverage values in each sample as the rest of the columns.

## Examples

```
data(f_tibble)

rel.cov = coverage_normalization(f_tibble)
```

---

DESeq2_l2fc  *Calculating log2 fold change for HTS-SIP data.*

---

## Description

The 'use_geo_mean' parameter uses geometric means on all non-zero abundances for estimateSize-Factors instead of using the default log-tranformed geometric means.

## Usage

```
DESeq2_l2fc(
  physeq,
  density_min,
  density_max,
  design,
  l2fc_threshold = 0.25,
  sparsity_threshold = 0.25,
  sparsity_apply = "all",
  size_factors = "geoMean"
)
```

## Arguments

| | |
|---|---|
| physeq | Phyloseq object |
| density_min | Minimum buoyant density of the 'heavy' gradient fractions |
| density_max | Maximum buoyant density of the 'heavy' gradient fractions |
| design | design parameter used for DESeq2 analysis. See DESeq2::DESeq for more details. |

l2fc_threshold  log2 fold change (l2fc) values must be significantly above this threshold in order
                to reject the hypothesis of equal counts.

sparsity_threshold

                All OTUs observed in less than this portion (fraction: 0-1) of gradient fraction
                samples are pruned. A a form of indepedent filtering, The sparsity cutoff with
                the most rejected hypotheses is used.

sparsity_apply  Apply sparsity threshold to all gradient fraction samples ('all') or just heavy
                fraction samples ('heavy')

size_factors    Method of estimating size factors. 'geoMean' is from (Pepe-Ranney et. al.,
                2016) and removes all zero-abundances from the calculation. 'default' is the
                default for estimateSizeFactors. 'iterate' is an alternative when every OTU has
                a zero in >=1 sample.

## Value

dataframe of HRSIP results

## Examples

```
data(phylo.qSIP)


df_l2fc = DESeq2_l2fc(phylo.qSIP, density_min=1.71, density_max=1.75, design=~Isotope)
```

---

df_atomX_boot                    *Bootstrapped atom fraction excess table*

---

## Description

Data table generated from bostrapping the AFE table using the "qSIP_bootstrap_fcr" function

## Usage

```
data(df_atomX_boot)
```

## Format

An object of class "data.frame"

---

filter_l2fc *Filter l2fc table*

---

### Description

filter_l2fc filters a l2fc table to 'best' sparsity cutoffs & bouyant density windows.

### Usage

```
filter_l2fc(df_l2fc, padj_cutoff = 0.1)
```

### Arguments

df_l2fc           data.frame of log2 fold change values

padj_cutoff      Adjusted p-value cutoff for rejecting the null hypothesis that l2fc values were not greater than the l2fc_threshold.

### Value

filtered df_l2fc object

---

filter_na *Remove MAGs with NAs from atomX table*

---

### Description

This function enables removing NAs from the atomX table.

### Usage

```
filter_na(atomX)
```

### Arguments

atomX           A list object created by qSIP_atom_excess_MAGs()

### Value

A list of 2 data.frame objects without MAGs which have NAs. 'W' contains the weighted mean buoyant density (W) values for each OTU in each treatment/control. 'A' contains the atom fraction excess values for each OTU. For the 'A' table, the 'Z' column is buoyant density shift, and the 'A' column is atom fraction excess.

## Examples

```
data(atomX)



### Remove NAs in atomX table
atomx_no_na = filter_na(atomX)
```

---

fractions                    *Fractions table*

---

### Description

Fractions data used for many functions in the package

### Usage

```
data(fractions)
```

### Format

An object of class "data.frame"

---

f_tibble                    *Coverage table*

---

### Description

Coverage data used for many functions in the package

### Usage

```
data(f_tibble)
```

### Format

An object of class "data.frame"

---

GC_content                        *GC_content table*

---

### Description

GC_content data

### Usage

```
data(GC_content)
```

### Format

An object of class "data.frame"

---

HRSIP                             *(MW-)HR-SIP analysis*

---

### Description

Conduct (multi-window) high resolution stable isotope probing (HR-SIP) analysis.

### Usage

```
HRSIP(
  physeq,
  design,
  density_windows = data.frame(density_min = c(1.7), density_max = c(1.75)),
  sparsity_threshold = seq(0, 0.3, 0.1),
  sparsity_apply = "all",
  l2fc_threshold = 0.25,
  padj_method = "BH",
  padj_cutoff = NULL,
  parallel = FALSE
)
```

### Arguments

physeq          Phyloseq object

design          design parameter used for DESeq2 analysis. This is usually used to differen-
                tiate labeled-treatment and unlabeld-control samples. See `DESeq2::DESeq` for
                more details on the option.

density_windows

                The buoyant density window(s) used for for calculating log2 fold change values.
                Input can be a vector (length 2) or a data.frame with a 'density_min' and a
                'density_max' column (each row designates a density window).

sparsity_threshold

> All OTUs observed in less than this portion (fraction: 0-1) of gradient fraction samples are pruned. This is a form of indepedent filtering. The sparsity cutoff with the most rejected hypotheses is used.

sparsity_apply   Apply sparsity threshold to all gradient fraction samples ('all') or just 'heavy' fraction samples ('heavy'), where 'heavy' samples are designated by the `density_windows`.

l2fc_threshold   log2 fold change (l2fc) values must be significantly above this threshold in order to reject the hypothesis of equal counts. See DESeq2 for more information.

padj_method      Method for global p-value adjustment (See `p.adjust()`).

padj_cutoff      Adjusted p-value cutoff for rejecting the null hypothesis that l2fc values were not greater than the l2fc_threshold. Set to `NULL` to skip filtering of results to the sparsity cutoff with most rejected hypotheses and filtering each OTU to the buoyant density window with the greatest log2 fold change.

parallel         Process each parameter combination in parallel. See `plyr::mdply()` for more information.

### Details

The (MW-)HR-SIP workflow is as follows:

1. For each sparsity threshold & BD window: calculate log2 fold change values (with DESeq2) for each OTU
2. Globally adjust p-values with a user-defined method (see p.adjust())
3. Select the sparsity cutoff with the most rejected hypotheses
4. For each OTU, select the BD window with the greatest log2 fold change value

### Value

dataframe of HRSIP results

### Examples

```
data(phylo.qSIP)


## HR-SIP
### Note: treatment-control samples differentiated with 'design=~Isotope'
df_l2fc = HRSIP(phylo.qSIP, design=~Isotope)
## Same, but multiple BD windows (MW-HR-SIP). For parallel processing change to parallel = TRUE
### Windows = 1.7-1.74, 1.72-1.75, and 1.73 - 1.76
windows = data.frame(density_min=c(1.71,1.72, 1.73), density_max=c(1.74,1.75,1.76))
df_l2fc = HRSIP(phylo.qSIP,
                design=~Isotope,
                density_windows = windows,
                padj_cutoff = 0.05,
                parallel=FALSE)
```

---

incorporators_taxonomy

*Isotope incorporator list with GTDB taxonomy*

---

## Description

This function provides a table with MAGs and their corresponding GTDB taxonomy as an output. This would be useful in identifying the taxa that have incorporation

## Usage

```
incorporators_taxonomy(taxonomy, bootstrapped_AFE_table)
```

## Arguments

taxonomy            A taxonomy tibble obtained in the markdown. This taxonomy tibble is typically a concatenated list of archaeal and bacterial taxonomy from GTDB-Tk Please check GTDB-Tk documentation for running the tool

bootstrapped_AFE_table

A data frame indicating bootstrapped atom fraction excess values

## Value

A tibble with two columns, OTU and Taxonomy, with taxonomy of the incorporator MAGs

## Examples

```
data(taxonomy_tibble,df_atomX_boot)



### Making incorporator taxonomy list
incorporator_list = incorporators_taxonomy(taxonomy = taxonomy_tibble,
        bootstrapped_AFE_table = df_atomX_boot)
```

---

mag.table            *MAG abundance table in phyloseq format*

---

## Description

MAG abundances in the format of phyloseq object to be used in the qSIP and (MW-)HR-SIP pipeline

**Usage**

```
data(mag.table)
```

**Format**

An object of class "phyloseq"

---

| phylo.qSIP | *Master phyloseq object* |
|---|---|

---

**Description**

Master phyloseq object

**Usage**

```
data(phylo.qSIP)
```

**Format**

An object of class "phyloseq"

---

| phylo.table | *Master phyloseq object using the MAG phyloseq objects* |
|---|---|

---

**Description**

Creates a phyloseq-style object using processed phyloseq objects for otu table (here, MAG table), taxa table, and sample table

**Usage**

```
phylo.table(mag, taxa, samples)
```

**Arguments**

| | |
|---|---|
| mag | phyloseq-styled MAG table |
| taxa | phyloseq-styled taxa table |
| samples | sample information table |

**Value**

phyloseq object for MAGs

## Examples

```
data(mag.table,taxonomy.object,samples.object,fractions,taxonomy_tibble)
###Making phyloseq table from fractions metadata
samples.object = sample.table(fractions)
taxonomy.object = tax.table(taxonomy_tibble)



### Making master phyloseq table from scaled MAG data, taxa and fractions phyloseq data
phylo.qSIP = phylo.table(mag.table,taxonomy.object,samples.object)
```

---

qSIP_atom_excess_format_MAGs

*Reformat a phyloseq object of qSIP_atom_excess_MAGs analysis*

---

## Description

Reformat a phyloseq object of qSIP_atom_excess_MAGs analysis

## Usage

```
qSIP_atom_excess_format_MAGs(physeq, control_expr, treatment_rep)
```

## Arguments

| | |
|---|---|
| physeq | A phyloseq object |
| control_expr | An expression for identifying unlabeled control samples in the phyloseq object (eg., "Substrate=='12C-Con'") |
| treatment_rep | Which column in the phyloseq sample data designates replicate treatments |

## Value

numeric value: atom fraction excess (A)

qSIP_atom_excess_MAGs    *Calculate atom fraction excess using q-SIP method*

**Description**

Calculate atom fraction excess using q-SIP method

**Usage**

```
qSIP_atom_excess_MAGs(
  physeq,
  control_expr,
  treatment_rep = NULL,
  isotope = "13C",
  df_OTU_W = NULL,
  Gi
)
```

**Arguments**

| | |
|---|---|
| physeq | A phyloseq object |
| control_expr | Expression used to identify control samples based on sample_data. |
| treatment_rep | Which column in the phyloseq sample data designates replicate treatments |
| isotope | The isotope for which the DNA is labeled with ('13C' or '18O') |
| df_OTU_W | Keep NULL |
| Gi | GC content of the MAG |

**Value**

A list of 2 data.frame objects. 'W' contains the weighted mean buoyant density (W) values for each OTU in each treatment/control. 'A' contains the atom fraction excess values for each OTU. For the 'A' table, the 'Z' column is buoyant density shift, and the 'A' column is atom fraction excess.

**Examples**

```
data(phylo.qSIP,GC_content)
### Making atomx table

## Not run::
### BD shift (Z) & atom excess (A)
atomX = qSIP_atom_excess_MAGs(phylo.qSIP,
                      control_expr='Isotope=="12C"',
                      treatment_rep='Replicate',
                      Gi = GC_content)
```

qSIP_bootstrap_fcr  *Calculate adjusted bootstrap CI after for multiple testing for atom fraction excess using q-SIP method. Multiple hypothesis tests are corrected by*

## Description

Calculate adjusted bootstrap CI after for multiple testing for atom fraction excess using q-SIP method. Multiple hypothesis tests are corrected by

## Usage

```
qSIP_bootstrap_fcr(
  atomX,
  isotope = "13C",
  n_sample = c(3, 3),
  ci_adjust_method = "fcr",
  n_boot = 10,
  parallel = FALSE,
  a = 0.1
)
```

## Arguments

atomX                A list object created by qSIP_atom_excess_MAGs()

isotope              The isotope for which the DNA is labeled with ('13C' or '18O')

n_sample             A vector of length 2. The sample size for data resampling (with replacement) for 1) control samples and 2) treatment samples.

ci_adjust_method
                     Confidence interval adjustment method. Please choose 'FCR', 'Bonferroni', or 'none' (if no adjustment is needed). Default is FCR and also provides unadjusted CI.

n_boot               Number of bootstrap replicates.

parallel             Parallel processing. See .parallel option in dplyr::mdply() for more details.

a                    A numeric value. The alpha for calculating confidence intervals.

## Value

A data.frame of atom fraction excess values (A) and atom fraction excess confidence intervals adjusted for multiple testing.

## Examples

```
data(phylo.qSIP,GC_content)

### BD shift (Z) & atom excess (A)
atomX = qSIP_atom_excess_MAGs(phylo.qSIP,
                        control_expr='Isotope=="12C"',
                        treatment_rep='Replicate', Gi = GC_content)

### Add doParallel::registerDoParallel(num_cores) if parallel bootstrapping is to be done
df_atomX_boot = qSIP_bootstrap_fcr(atomX, n_boot=5, parallel = FALSE)
```

---

| sample.table | *phyloseq-styled sample table* |
|---|---|

---

## Description

Creates a phyloseq-styled sample table from fractions metadata containing data on fraction number, number of replicates, buoyant density calculated from a refractometer, type of isotope, and DNA concentration of each fraction, and isotope type. See below for information on "fractions" file.

## Usage

```
sample.table(fractions_df)
```

## Arguments

fractions_df        fractions data frame A fractions file with the following columns

- Replicate: Depends on how many replicates the study has
- Fractions: Typically in the range of 2-24
- Buoyant_density: As calculated from the refractometer for each fraction and replicate
- Isotope: "12C", "13C", "14N", "15N" etc.
- DNA_concentration
- Sample: In the format "'isotope'*rep*#*fraction*#". For instance, "12C_rep_1_fraction_1"

## Value

data frame: phyloseq-style sample table

## Examples

```
data(fractions)



### Making phyloseq table from fractions metadata
samples.object = sample.table(fractions)
```

---

samples.object          *Fractions table in phyloseq format*

---

## Description

Fractions metadata in the format of phyloseq object to be used in the qSIP and (MW-)HR-SIP pipeline

## Usage

```
data(samples.object)
```

## Format

An object of class "phyloseq"

---

scale_features_lm          *Scale feature coverage values to estimate their absolute abundance*

---

## Description

Calculates global scaling factors for features (contigs or bins),based on linear regression of sequin coverage. Options include log-transformations of coverage, as well as filtering features based on limit of detection. This function must be called first, before the feature abundance table, feature detection table, and plots are retrieved.

## Usage

```
scale_features_lm(
  f_tibble,
  sequin_meta,
  seq_dilution,
  log_trans = TRUE,
  coe_of_variation = 250,
  lod_limit = 0,
  save_plots = TRUE,
  plot_dir = tempdir(),
  cook_filtering = TRUE
)
```

**Arguments**

| | |
|---|---|
| f_tibble | Can be either of (1) a tibble with first column "Feature" that contains bin IDs, and the rest of the columns represent samples with bins' coverage values. (2) a tibble as outputted by the program "checkm coverage" from the tool CheckM. Please check CheckM documentation - https://github.com/Ecogenomics/CheckM on the usage for "checkm coverage" program |
| sequin_meta | tibble containing sequin names ("Feature column") and concentrations in atta-moles/uL ("Concentration") column. |
| seq_dilution | tibble with first column "Sample" with **same sample names as in f_tibble**, and a second column "Dilution" showing ratio of sequins added to final sample volume (e.g. a value of 0.01 for a dilution of 1 volume sequin to 99 volumes sample) |
| log_trans | Boolean (TRUE or FALSE), should coverages and sequin concentrations be log-scaled? |
| coe_of_variation | |
| | Acceptable coefficient of variation for coverage and detection (eg. 20 - for 20 % threshold of coefficient of variation). Coverages above the threshold value will be flagged in the plots. |
| lod_limit | (Decimal range 0-1) Threshold for the percentage of minimum detected sequins per concentration group. Default = 0 |
| save_plots | Boolean (TRUE or FALSE), should sequin scaling be saved? Default = TRUE |
| plot_dir | Directory where plots are to be saved. Will create a directory "sequin_scaling_plots_lm" if it does not exist. |
| cook_filtering | Boolean (TRUE or FALSE), should data points be filtered based on Cook's distance metric. Cooks distance can be useful in detecting influential outliers in an ordinary least square's regression model, which can negatively influence the model. A threshold of Cooks distance of 4/n (where n is the sample size) is chosen, and any data point with Cooks distance > 4/n is filtered out. It is typical to choose 4/n as the threshold in detecting the outliers in the data. Default = TRUE |

**Value**

a list of tibbles containing

- mag_tab: a tibble with first column "Feature" that contains bin (or contig IDs), and the rest of the columns represent samples with features' scaled abundances (attamoles/uL)

- mag_det: a tibble with first column "Feature" that contains bin (or contig IDs),

- plots: linear regression plots for scaling MAG coverage values to absolute abundance

- scale_fac: a master tibble with all of the intermediate values in above calculations

**Examples**

```
data(f_tibble, sequins, seq_dil)
```

```
### scaling sequins from coverage values
scaled_features_lm = scale_features_lm(f_tibble,sequin_meta, seq_dil)
```

---

scale_features_rlm           *Scale feature coverage values to estimate their absolute abundance*

---

#### Description

Calculates global scaling factors for features (contigs or bins),based on linear regression of sequin coverage. Options include log-transformations of coverage, as well as filtering features based on limit of detection. This function must be called first, before the feature abundance table, feature detection table, and plots are retrieved.

#### Usage

```
scale_features_rlm(
  f_tibble,
  sequin_meta,
  seq_dilution,
  log_trans = TRUE,
  coe_of_variation = 250,
  lod_limit = 0,
  save_plots = TRUE,
  plot_dir = tempdir()
)
```

#### Arguments

| | |
|---|---|
| f_tibble | Can be either of (1) a tibble with first column "Feature" that contains bin IDs, and the rest of the columns represent samples with bins' coverage values. (2) a tibble as outputted by the program "checkm coverage" from the tool CheckM. Please check CheckM documentation - https://github.com/Ecogenomics/CheckM on the usage for "checkm coverage" program |
| sequin_meta | tibble containing sequin names ("Feature column") and concentrations in atta-moles/uL ("Concentration") column. |
| seq_dilution | tibble with first column "Sample" with **same sample names as in f_tibble**, and a second column "Dilution" showing ratio of sequins added to final sample volume (e.g. a value of 0.01 for a dilution of 1 volume sequin to 99 volumes sample) |
| log_trans | Boolean (TRUE or FALSE), should coverages and sequin concentrations be log-scaled? Default = TRUE |
| coe_of_variation | |
| | Acceptable coefficient of variation for coverage and detection (eg. 20 - for 20 % threshold of coefficient of variation). Coverages above the threshold value will be flagged in the plots. Default = 250 |

| lod_limit | (Decimal range 0-1) Threshold for the percentage of minimum detected sequins per concentration group. Default = 0 |
|---|---|
| save_plots | Boolean (TRUE or FALSE), should sequin scaling be saved? Default = TRUE |
| plot_dir | Directory where plots are to be saved. Will create a directory "sequin_scaling_plots_rlm" if it does not exist. |

## Value

a list of tibbles containing

- mag_tab: a tibble with first column "Feature" that contains bin (or contig IDs), and the rest of the columns represent samples with features' scaled abundances (attamoles/uL)

- mag_det: a tibble with first column "Feature" that contains bin (or contig IDs),

- plots: linear regression plots for scaling MAG coverage values to absolute abundance (optional)

- scale_fac: a master tibble with all of the intermediate values in above calculations

## Examples

```
data(f_tibble, sequins, seq_dil)



### scaling sequins from coverage values
scaled_features_rlm = scale_features_rlm(f_tibble,sequins, seq_dil)
```

---

| sequins | *Sequins table* |
|---|---|

---

## Description

Sequins metadata

## Usage

```
data(sequins)
```

## Format

An object of class "data.frame"

| seq_dil | *Sequins dilution table* |
|---|---|

### Description

Sequins dilution data

### Usage

```
data(seq_dil)
```

### Format

An object of class "data.frame"

| tax.table | *phyloseq taxa table from GTDB taxonomy input* |
|---|---|

### Description

A MAG table, similar to OTU table in phyloseq, will be generated from a concantenated GTDB taxa table for bacteria and archaea

### Usage

```
tax.table(taxonomy)
```

### Arguments

taxonomy       GTDB taxonomy data frame. A taxonomy file in the GTDB output format. Load the bacteria and archaea taxonomy outputs separately. The markdown requires loading the standard output files from GTDB-Tk separately for bacteria and archaea

### Value

phyloseq-style taxonomy table, but for MAGs

### Examples

```
data(taxonomy_tibble)


### Making phyloseq table from taxonomy metadata
taxonomy.object = tax.table(taxonomy_tibble)
```

---

taxonomy.object            *Taxonomy table in phyloseq format*

---

### Description

Taxonomy table in the format of phyloseq object to be used in the qSIP and (MW-)HR-SIP pipeline

### Usage

```
data(taxonomy.object)
```

### Format

An object of class "phyloseq"

---

taxonomy_tibble            *Taxonomy table*

---

### Description

Taxonomy table from GTDB-Tk output - combining both bacterial and archaeal taxonomy

### Usage

```
data(taxonomy_tibble)
```

### Format

An object of class "data.frame"

# Index