

Package: cuperdec (via r-universe)

October 13, 2024

Type Package

Title Cumulative Percent Decay Curve Generator

Version 1.1.0.9000

Description Calculates and visualises cumulative percent 'decay' curves, which are typically calculated from metagenomic taxonomic profiles. These can be used to estimate the level of expected 'endogenous' taxa at different abundance levels retrieved from metagenomic samples, when comparing to samples of known sampling site or source. Method described in Fellows Yates, J. A. et. al. (2021) Proceedings of the National Academy of Sciences USA <[doi:10.1073/pnas.2021655118](https://doi.org/10.1073/pnas.2021655118)>.

License MIT + file LICENSE

URL <https://github.com/jfy133/cuperdec>

BugReports <https://github.com/jfy133/cuperdec/issues>

Depends R (>= 3.5.0)

Imports dplyr, ggplot2, magrittr, readr, rlang, tidyr

Suggests knitr, rmarkdown, testthat, tibble

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

LazyData true

NeedsCompilation no

RoxygenNote 7.1.2

Repository <https://jfy133.r-universe.dev>

RemoteUrl <https://github.com/jfy133/cuperdec>

RemoteRef HEAD

RemoteSha b06243e629983f76e4d24d493ac706fe67f6cd5b

Contents

adaptive_burnin_filter	2
calculate_curve	3
cuperdec_database_ex	3
cuperdec_metadata_ex	4
cuperdec_taxatable_ex	5
hard_burnin_filter	5
load_database	6
load_map	7
load_taxa_table	8
plot_cuperdec	8
simple_filter	9

Index	11
--------------	-----------

adaptive_burnin_filter

Calculate adaptive burn-in retain/discard list

Description

Automates a selection of a per-sample 'burn in' based on the nature of the sample's curve itself (rather than supplying a hard value) by finding the point from which the 'fluctuation' of the curve doesn't exceed the mean +/- SD of the total curve.

Usage

```
adaptive_burnin_filter(curves, percent_threshold)
```

Arguments

`curves` A cuperdec curve table calculated with `calculate_curve`.

`percent_threshold` A percentage of the target-source in a sample above which a sample is considered 'retained'.

Value

A tibble with each row showing each sample and whether it passed the specified filter.

Examples

```
data(cuperdec_taxatable_ex)
data(cuperdec_database_ex)

taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")
```

```
curve_results <- calculate_curve(taxa_table, iso_database)
adaptive_burnin_filter(curve_results, percent_threshold = 0.1)
```

calculate_curve	<i>Calculate cumulative decay percent curve</i>
-----------------	---

Description

Performs the initial decay curve based on percentage of 'target' isolation source along a rank of most to least abundant taxa for a given sample.

Usage

```
calculate_curve(taxa_table, database)
```

Arguments

taxa_table	An OTU table loaded with load_taxa_table .
database	A database file loaded with load_database .

Value

An object in the form of a tibble with taxa of each given sample ordered by rank and the proportion of taxa up to that rank deriving from your target source.

Examples

```
data(cuperdec_taxatable_ex)
data(cuperdec_database_ex)

taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")

calculate_curve(taxa_table, iso_database)
```

cuperdec_database_ex	<i>Example isolation source database input for cuperdec</i>
----------------------	---

Description

Example isolation source database used for input to cuperdec based. Species names are from a NCBI Nt database and isolation sources gather from the Human Oral Microbiome database, NCBI GenBank, and manual curation.

Usage

```
data(cuperdec_database_ex)
```

Format

An TSV table loaded as a tibble.

Source

[doi:10.5281/zenodo.3740492](https://doi.org/10.5281/zenodo.3740492)

Examples

```
data(cuperdec_database_ex)
load_database(cuperdec_database_ex, target = "oral")
```

cuperdec_metadata_ex *Example metadata file input for cuperdec*

Description

Example metadata map file corresponding to samples in example data "cuperdec_taxatable_ex". Includes a grouping column corresponding to sample species.

Usage

```
data(cuperdec_metadata_ex)
```

Format

An TSV table loaded as a tibble.

Source

[doi:10.5281/zenodo.3740492](https://doi.org/10.5281/zenodo.3740492)

Examples

```
data(cuperdec_metadata_ex)
load_map(cuperdec_metadata_ex, sample_col = "#SampleID", source_col = "Env")
```

cuperdec_taxatable_ex *Example taxon table input for cuperdec*

Description

Example taxon table used for input to cuperdec based on data including shotgun-sequenced ancient calculus samples aligned against the NCBI Nt database from Oct 2017 using MALT. Samples are columns, rows are taxa and counts are assigned reads.

Usage

```
data(cuperdec_taxatable_ex)
```

Format

An TSV table loaded as a tibble.

Source

[doi:10.5281/zenodo.3740492](https://doi.org/10.5281/zenodo.3740492)

Examples

```
data(cuperdec_taxatable_ex)
load_taxa_table(cuperdec_taxatable_ex)
```

hard_burnin_filter *Calculate hard burn-in retain/discard list*

Description

Returns a table of whether each sample passes a given threshold, after considering a 'burn-in', in the form of a fraction of the abundance ranks.

Usage

```
hard_burnin_filter(curves, percent_threshold, rank_burnin)
```

Arguments

curves	A cuperdec curve table calculated with calculate_curve .
percent_threshold	A percentage of the target-source in a sample above which a sample is considered 'retained'.
rank_burnin	A number between 0 and 1 indicating the fraction of taxa to ignore before applying the threshold.

Value

A tibble with each row showing each sample and whether it passed the specified filter.

Examples

```
data(cuperdec_taxatable_ex)
data(cuperdec_database_ex)

taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")

curve_results <- calculate_curve(taxa_table, iso_database)
hard_burnin_filter(curve_results, percent_threshold = 50, rank_burnin = 0.1)
```

load_database	<i>Load database</i>
---------------	----------------------

Description

Loads a taxon/isolation source database file, i.e. first column is a list of taxa, and the second column is a list of isolation sources, and formats for downstream analysis.

Usage

```
load_database(x, target)
```

Arguments

x	Path to a (minimum) two column TSV file or tidy dataframe (e.g. tibble), one column with taxon names and other indicating if from target isolation source.
target	the string in the 'Isolation Source' (i.e. 2nd) column which is the expected target source of the samples

Details

Taxon names should match that with the taxa table.

Value

A tibble, formatted for use in downstream cuperdec functions.

Examples

```
data(cuperdec_database_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")
```

load_map	<i>Load metadata table</i>
----------	----------------------------

Description

Loads a metadata table and reformats it for downstream analysis. This needs to include at minimum two columns: sample name, and sample source.

Usage

```
load_map(x, sample_col, source_col)
```

Arguments

x	Path to a TSV file or tidy dataframe (e.g. tibble) with a column containing sample names and other grouping metadata columns.
sample_col	A column name specifying which column should be used to specify sample names.
source_col	A column name specifying which group or the source the sample is from.

Details

The two columns required need to include the following information:

- Sample name - a unique identifier for each sample
- Sample source - a grouping ID indicating what 'source' the sample is from This is used for plotting to separate comparative 'sources' to your own samples.

Value

A tibble, formatted for use in downstream cuperdec functions.

Examples

```
data(cuperdec_metadata_ex)
metadata_table <- load_map(cuperdec_metadata_ex,
  sample_col = "#SampleID",
  source_col = "Env"
)
```

load_taxa_table	<i>Load OTU table</i>
-----------------	-----------------------

Description

Loads a typical taxa table (Samples: columns; Taxa: rows) in TSV format and standardises some columns, storing the table in the form of a tibble.

Usage

```
load_taxa_table(x)
```

Arguments

x Path to a TSV file or tidy dataframe (e.g. tibble) consisting of an OTU table of samples as columns, except first column with taxon names.

Value

A tibble, formatted for use in downstream cuperdec functions.

Examples

```
data(cuperdec_taxatable_ex)
taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
```

plot_cuperdec	<i>Plot cumulative percent decay curves</i>
---------------	---

Description

Generates visual representation of curves, with optional separate plotting of different groups, and also indication of individuals passing different on types filters.

Usage

```
plot_cuperdec(
  curves,
  metadata,
  burnin_result,
  restrict_x = 0,
  facet_cols = NULL
)
```


Arguments

curves	Output tibble from calculate_curve .
metadata	Output from load_map .
burnin_result	Output from apply_*_burnin. functions (optional).
restrict_x	Restrict viewing of abundance rank to X number of ranks (useful for closer inspection of curves) (optional).
facet_cols	Custom number of columns for faceted plots (optional).

Value

A ggplot2 image object.

Examples

```

data(cuperdec_taxatable_ex)
data(cuperdec_database_ex)
data(cuperdec_metadata_ex)

taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")
metadata_table <- load_map(cuperdec_metadata_ex,
  sample_col = "#SampleID",
  source_col = "Env"
)

curves <- calculate_curve(taxa_table, iso_database)
burnin_results <- adaptive_burnin_filter(curves, percent_threshold = 0.1)

plot_cuperdec(curves, metadata_table, burnin_results)

```

simple_filter *Apply simple percentage filter*

Description

Performs the initial decay curve based on percentage of 'target' isolation source along a rank of most to least abundant taxa for a given sample.

Usage

```
simple_filter(curves, percent_threshold)
```

Arguments

curves	A cuperdec curve table calculated with calculate_curve .
percent_threshold	A database file loaded with load_database .

Value

A tibble with each row showing each sample and whether it passed the specified filter.

Examples

```
data(cuperdec_taxatable_ex)
data(cuperdec_database_ex)

taxa_table <- load_taxa_table(cuperdec_taxatable_ex)
iso_database <- load_database(cuperdec_database_ex, target = "oral")

curve_results <- calculate_curve(taxa_table, iso_database)
simple_filter(curve_results, percent_threshold = 50)
```

Index

* datasets

- cuperdec_database_ex, [3](#)
- cuperdec_metadata_ex, [4](#)
- cuperdec_taxatable_ex, [5](#)

adaptive_burnin_filter, [2](#)

calculate_curve, [2](#), [3](#), [5](#), [9](#)
cuperdec_database_ex, [3](#)
cuperdec_metadata_ex, [4](#)
cuperdec_taxatable_ex, [5](#)

hard_burnin_filter, [5](#)

load_database, [3](#), [6](#), [9](#)
load_map, [7](#), [9](#)
load_taxa_table, [3](#), [8](#)

plot_cuperdec, [8](#)

simple_filter, [9](#)