Package: dbpq (via r-universe)

May 15, 2026

Title Manage FASTA Reference Databases for Taxonomic Assignment

Version 0.0.0.9000

Description Download, format, summarize, and modify FASTA reference databases used for taxonomic assignment in metabarcoding pipelines. Supports major databases (UNITE, SILVA, PR2, BOLD, MaarjAM, Eukaryome) and converts between taxonomy header formats (dada2, SINTAX). Part of the 'pqverse' ecosystem.

License MIT + file LICENSE

URL https://github.com/adrientaudiere/dbpq

BugReports https://github.com/adrientaudiere/dbpq/issues

Depends R (>= 4.1.0)

Imports Biostrings, dplyr, purrr, stringr, tibble, tidyr, tools

Suggests dada2, knitr, rmarkdown, R.utils, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev zlib1g-dev

Repository https://adrientaudiere.r-universe.dev

Date/Publication 2026-03-19 10:37:47 UTC

RemoteUrl https://github.com/adrientaudiere/dbpq

RemoteRef HEAD

RemoteSha 9ddef4490b11c56dca14dcd73d27a51e96233a99

Contents

`count_pattern_db` *Count lines matching a pattern in a FASTA file*

Description

Count lines (sequences if fasta file) matching a pattern. Accepts gzip files. May not work on Windows.

Usage

```
count_pattern_db(file, pattern = ">")
```

Arguments

`file`	(Character, required) Path to a file (plain or gzip), often a FASTA file.
`pattern`	(Character, default `">"`) A pattern to search for.

Value

An integer, the number of matching lines.

Author(s)

Adrien Taudière

See Also

`filter_db()`, `count_seq_db()`

Examples

```
# count_pattern_db("my_database.fasta", "Fungi")
```

count_seq_db *Count sequences in a FASTA file*

Description

Counts the number of sequences in a FASTA file by counting header lines (lines starting with >). Accepts gzip files.

Usage

```
count_seq_db(file)
```

Arguments

file	(Character, required) Path to a FASTA file (plain or gzip).

Value

An integer, the number of sequences.

Author(s)

Adrien Taudière

See Also

count_pattern_db()

Examples

```
# count_seq_db("my_database.fasta")
```

cutadapt_rm_primers_db *Remove primers from a FASTA database using cutadapt*

Description

Removes pairs of primers and flanking regions from a FASTA reference database using cutadapt. Uses linked adapters to trim between forward and reverse primers.

Usage

```
cutadapt_rm_primers_db(
  ref_fasta,
  output = NULL,
  primer_fw = NULL,
  primer_rev = NULL,
  discard_untrimmed = TRUE,
  nproc = 1,
  verbose = TRUE,
  cmd_is_run = TRUE,
  return_file_path = FALSE,
  start_with_fw = FALSE,
  output_json = FALSE,
  error_tolerance = 0.1,
  args_before_cutadapt = paste0("source ~/miniforge3/etc/profile.d/conda.sh ",
    "&& conda activate cutadaptenv && ")
)
```

Arguments

`ref_fasta`	(Character, required) Path to a FASTA file (plain or gzip).
`output`	(Character) Path to the output FASTA file. If NULL, defaults to `{basename}_cutadapted.fasta`.
`primer_fw`	(Character, required) The forward primer DNA sequence.
`primer_rev`	(Character, required) The reverse primer DNA sequence.
`discard_untrimmed`	(Logical, default `TRUE`) Discard sequences where primers were not found.
`nproc`	(Integer, default `1`) Number of CPU cores for cutadapt.
`verbose`	(Logical, default `TRUE`) Print summary statistics.
`cmd_is_run`	(Logical, default `TRUE`) If FALSE, return the command string without executing it.
`return_file_path`	(Logical, default `FALSE`) If TRUE, return the output file path instead of the command.
`start_with_fw`	(Logical, default `FALSE`) If TRUE, the forward primer must be anchored at the start of the sequence.
`output_json`	(Logical, default `FALSE`) If TRUE, write a JSON summary of the cutadapt process.
`error_tolerance`	(Numeric, default `0.1`) Maximum error rate for primer matching.
`args_before_cutadapt`	(Character) Shell commands to run before cutadapt (e.g., conda activation).

Details

This function is mainly a wrapper of the work of others. Please cite cutadapt (doi:10.14806/ej.17.1.200).

Value

The cutadapt command string, or the output file path if `return_file_path = TRUE`.

Author(s)

Adrien Taudière

Examples

```
## Not run:
cutadapt_rm_primers_db(
  "database.fasta.gz",
  output = "db_cutadapted.fasta",
  primer_fw = "GCATCGATGAAGAACGCAGC",
  primer_rev = "TCCTCCGCTTATTGATATGC"
)

## End(Not run)
```

`download_bold_db` *Download a BOLD reference database*

Description

Downloads reference sequences from BOLD Systems (Barcode of Life Data).

Usage

```
download_bold_db(dest_dir = ".", marker = "COI-5P", verbose = TRUE)
```

Arguments

`dest_dir`	(Character, default `"."`) Directory to save the downloaded file.
`marker`	(Character, default `"COI-5P"`) The barcode marker to download.
`verbose`	(Logical, default `TRUE`) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_bold_db()

## End(Not run)
```

`download_eukaryome_db` *Download the Eukaryome reference database*

Description

Downloads the Eukaryome database.

Usage

```
download_eukaryome_db(dest_dir = ".", verbose = TRUE)
```

Arguments

`dest_dir`	(Character, default `"."`) Directory to save the downloaded file.
`verbose`	(Logical, default `TRUE`) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_eukaryome_db()

## End(Not run)
```

download_marjaam_db *Download the MaarjAM reference database*

Description

Downloads the MaarjAM database for arbuscular mycorrhizal fungi (AMF).

Usage

```
download_marjaam_db(dest_dir = ".", verbose = TRUE)
```

Arguments

dest_dir	(Character, default ".") Directory to save the downloaded file.
verbose	(Logical, default TRUE) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_marjaam_db()

## End(Not run)
```

download_pr2_db *Download a PR2 reference database*

Description

Downloads the PR2 protist ribosomal reference database.

Usage

```
download_pr2_db(dest_dir = ".", version = NULL, verbose = TRUE)
```

Arguments

dest_dir	(Character, default ".") Directory to save the downloaded file.
version	(Character) PR2 version number.
verbose	(Logical, default TRUE) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_pr2_db()

## End(Not run)
```

`download_silva_db` *Download a SILVA reference database*

Description

Downloads the SILVA ribosomal RNA database (16S/18S).

Usage

```
download_silva_db(
  dest_dir = ".",
  version = NULL,
  target = c("SSU", "LSU"),
  verbose = TRUE
)
```

Arguments

`dest_dir`	(Character, default `"."`) Directory to save the downloaded file.
`version`	(Character) SILVA version number (e.g., `"138.2"`).
`target`	(Character, default `"SSU"`) One of `"SSU"` or `"LSU"`.
`verbose`	(Logical, default `TRUE`) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_silva_db()

## End(Not run)
```

`download_unite_db` *Download a UNITE reference database*

Description

Downloads the latest UNITE fungal ITS database for taxonomic assignment.

Usage

```
download_unite_db(
  dest_dir = ".",
  type = c("dynamic", "static"),
  taxon_group = c("fungi", "eukaryotes"),
  verbose = TRUE
)
```

Arguments

`dest_dir`	(Character, default `"."`) Directory to save the downloaded file.
`type`	(Character, default `"dynamic"`) One of `"dynamic"` or `"static"`. Dynamic files include singletons, static do not.
`taxon_group`	(Character, default `"fungi"`) One of `"fungi"` or `"eukaryotes"`.
`verbose`	(Logical, default `TRUE`) Print progress messages.

Value

The path to the downloaded file (invisibly).

Author(s)

Adrien Taudière

Examples

```
## Not run:
download_unite_db()

## End(Not run)
```

`filter_db` *Filter a FASTA database by taxonomic pattern*

Description

Filters sequences from a FASTA database whose header lines match a given pattern. Accepts gzip files. May not work on Windows.

Usage

```
filter_db(
  ref_fasta,
  pattern,
  output = NULL,
  force_two_lines_per_seq = TRUE,
  keep_temporary_files = FALSE
)
```

Arguments

`ref_fasta`	(Character, required) Path to a FASTA file (plain or gzip).
`pattern`	(Character, required) A pattern to search for in sequence headers.
`output`	(Character, required) Path to the output FASTA file (must not be gzipped).
`force_two_lines_per_seq`	(Logical, default `TRUE`) Force the FASTA file to have exactly two lines per sequence (one header, one nucleotide line). If `FALSE`, the input must already be in this format.
`keep_temporary_files`	(Logical, default `FALSE`) If `TRUE` and `force_two_lines_per_seq` is `TRUE`, keep intermediate temporary files.

Value

The path to the output file (invisibly).

Author(s)

Adrien Taudière

See Also

`count_pattern_db()`

Examples

```
# filter_db("database.fasta.gz", "Rhizophydiales", "output.fasta")
```

format2dada2 *Format taxonomy headers for dada2::assignTaxonomy*

Description

Converts taxonomy headers to the format expected by `dada2::assignTaxonomy()`: `Kingdom;Phylum;Class;Order;Family`

Usage

```
format2dada2(
  fasta_db = NULL,
  taxnames = NULL,
  output_path = NULL,
  from_sintax = TRUE,
  pattern_to_remove = NULL,
  ...
)
```

Arguments

`fasta_db`	(Character) Path to a FASTA file. Mutually exclusive with `taxnames`.
`taxnames`	(Character vector) Taxonomy header strings. Mutually exclusive with `fasta_db`.
`output_path`	(Character) If provided and `fasta_db` is used, write the reformatted FASTA to this path.
`from_sintax`	(Logical, default `TRUE`) If TRUE, input is in SINTAX format. If FALSE, input is converted from standard format via `format2sintax()` first.
`pattern_to_remove`	(Character) Optional regex pattern to remove from the reformatted names.
`...`	Additional arguments passed to `format2sintax()` when `from_sintax = FALSE`.

Value

If `taxnames` is used, a character vector. If `fasta_db` is used, a `DNAStringSet` with reformatted names. When `output_path` is provided, the FASTA file is written and the `DNAStringSet` is returned invisibly.

Author(s)

Adrien Taudière

See Also

`format2sintax()`, `format2dada2_species()`

Examples

```
format2dada2(
  taxnames = "AB123;tax=k:Fungi,p:Ascomycota,c:Sordariomycetes",
  from_sintax = TRUE
)
```

`format2dada2_species` *Format taxonomy headers for dada2::addSpecies*

Description

Converts taxonomy headers to the format expected by `dada2::addSpecies()`: `AccessionID Genus Species`.

Usage

```
format2dada2_species(
  fasta_db = NULL,
  taxnames = NULL,
  from_sintax = FALSE,
  output_path = NULL,
  ...
)
```

Arguments

`fasta_db`	(Character) Path to a FASTA file. Mutually exclusive with `taxnames`.
`taxnames`	(Character vector) Taxonomy header strings. Mutually exclusive with `fasta_db`.
`from_sintax`	(Logical, default `FALSE`) If TRUE, input is in SINTAX format. If FALSE, input uses standard `k__` format.
`output_path`	(Character) If provided and `fasta_db` is used, write the reformatted FASTA to this path.
`...`	Additional arguments passed to internal functions.

Value

If `taxnames` is used, a character vector. If `fasta_db` is used, a `DNAStringSet` with reformatted names.

Author(s)

Adrien Taudière

See Also

`format2dada2()`, `format2sintax()`

Examples

```
format2dada2_species(
  taxnames = "AB123;k__Fungi;g__Aspergillus;s__fumigatus",
  from_sintax = FALSE
)
```

`format2sintax` *Format taxonomy headers to SINTAX format*

Description

Converts taxonomy headers from the common `k__Kingdom;p__Phylum;...` format to the VSEARCH SINTAX format (`tax=k:Kingdom,p:Phylum,...`).

Usage

```
format2sintax(
  fasta_db = NULL,
  taxnames = NULL,
  pattern_tax = "k__",
  pattern_sintax = "tax=k:",
  output_path = NULL
)
```

Arguments

`fasta_db`	(Character) Path to a FASTA file. Mutually exclusive with `taxnames`.
`taxnames`	(Character vector) Taxonomy header strings. Mutually exclusive with `fasta_db`.
`pattern_tax`	(Character, default `"k__"`) Pattern identifying the start of taxonomy in the original format.
`pattern_sintax`	(Character, default `"tax=k:"`) Pattern for the start of taxonomy in SINTAX format.
`output_path`	(Character) If provided and `fasta_db` is used, write the reformatted FASTA to this path.

Value

If `taxnames` is used, a character vector of reformatted names. If `fasta_db` is used, a `DNAStringSet` with reformatted names.

Author(s)

Adrien Taudière

See Also

`format2dada2()`, `format2dada2_species()`

Examples

```
format2sintax(taxnames = "AB123;k__Fungi;p__Ascomycota;c__Sordariomycetes")
```

---

get_file_extension *Get file extension(s)*

Description

Get file extension(s)

Usage

```
get_file_extension(file_path)
```

Arguments

file_path	(Character, required) Path to a file.

Value

A character vector of file extensions.

Examples

```
get_file_extension("my_database.fasta")
get_file_extension("my_database.fasta.gz")
```

---

list_ranks_db *List and count taxonomic ranks from a FASTA database*

Description

Extracts and counts occurrences of a given taxonomic rank from FASTA sequence headers. Requires taxonomy encoded in headers following the convention `k__Kingdom;p__Phylum;...` or similar.

Usage

```
list_ranks_db(file, rank_prefix = "k__")
```

Arguments

file	(Character, required) Path to a FASTA file (plain or gzip).
rank_prefix	(Character, default `"k__"`) The prefix identifying the taxonomic rank to extract (e.g., `"k__"` for kingdom, `"p__"` for phylum, `"c__"` for class, `"o__"` for order, `"f__"` for family, `"g__"` for genus, `"s__"` for species).

Value

A named integer vector of counts, sorted in decreasing order. Names are the taxonomic rank values.

Author(s)

Adrien Taudière

Examples

```
# list_ranks_db("my_database.fasta", rank_prefix = "p__")
```

`summarize_db` *Summarize a FASTA reference database*

Description

Provides an overview of a FASTA reference database: number of sequences, sequence length distribution, and taxonomic coverage at each rank.

Usage

```
summarize_db(
  file,
  rank_prefixes = c("k__", "p__", "c__", "o__", "f__", "g__", "s__")
)
```

Arguments

`file`	(Character, required) Path to a FASTA file (plain or gzip).
`rank_prefixes`	(Character vector) Taxonomic rank prefixes to summarize. Defaults to kingdom through species.

Value

A list with components:

- `n_sequences`: total number of sequences
- `length_summary`: summary statistics of sequence lengths
- `ranks`: a named list of unique count per rank

Author(s)

Adrien Taudière

Examples

```
# summarize_db("my_database.fasta")
```

Index