Package 'demulticoder'

May 5, 2025

Title Simultaneous Analysis of Multiplexed Metabarcodes

Version 0.1.2

Description A comprehensive set of wrapper functions for the analysis of multiplex metabarcode data. It includes robust wrappers for 'Cutadapt' and 'DADA2' to trim primers, filter reads, perform amplicon sequence variant (ASV) inference, and assign taxonomy. The package can handle single metabarcode datasets, datasets with two pooled metabarcodes, or multiple datasets simultaneously. The final output is a matrix per metabarcode, containing both ASV abundance data and associated taxonomic assignments. An optional function converts these matrices into 'phyloseq' and 'taxmap' objects. For more information on 'DADA2', including information on how DADA2 infers samples sequences, see Callahan et al. (2016) <doi:10.1038/nmeth.3869>. For more details on the demulticoder R package see Sudermann et al. (2025) <doi:10.1094/PHYTO-02-25-0043-FI>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.0.2)

Imports furrr, purrr, readr, stringr, tidyr, dplyr, ggplot2, tibble, utils

Suggests BiocManager, Biostrings, dada2, metacoder, ShortRead, phyloseq, rmarkdown, RcppParallel, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://grunwaldlab.github.io/demulticoder/, https://github.com/grunwaldlab/demulticoder

BugReports https://github.com/grunwaldlab/demulticoder/issues

RoxygenNote 7.3.2

Config/Needs/website rmarkdown

NeedsCompilation no

Maintainer Martha A. Sudermann <sudermam@oregonstate.edu>

Author Martha A. Sudermann [aut, cre, cph], Zachary S. L Foster [aut], Samantha Dawson [aut],

Hung Phan [aut],
Jeff H. Chang [aut],
Niklaus Grünwald [aut, cph]

Repository CRAN

Date/Publication 2025-05-05 09:50:02 UTC

Contents

assign_tax 2
convert_asv_matrix_to_objs 4
cut_trim 5
make_asv_abund_matrix 6
prepare_reads 7

Index **9**

`assign_tax` *Assign taxonomy functions*

Description

Assign taxonomy functions

Usage

```
assign_tax(
  analysis_setup,
  asv_abund_matrix,
  retrieve_files = FALSE,
  overwrite_existing = FALSE,
  db_rps10 = "oomycetedb.fasta",
  db_its = "fungidb.fasta",
  db_16S = "bacteriadb.fasta",
  db_other1 = "otherdb1.fasta",
  db_other2 = "otherdb2.fasta"
)
```

Arguments

`analysis_setup`	An object containing directory paths and data tables, produced by the `prepare_reads` function
`asv_abund_matrix`	The final abundance matrix containing amplified sequence variants
`retrieve_files`	Logical, TRUE/FALSE whether to copy files from the temp directory to the output directory. Default is FALSE.

`overwrite_existing`
: Logical, indicating whether to remove or overwrite existing files and directories from previous runs. Default is `FALSE`.

`db_rps10`
: The reference database for the rps10 metabarcode

`db_its`
: The reference database for the ITS metabarcode

`db_16S`
: The SILVA 16S-rRNA reference database provided by the user

`db_other1`
: The reference database for other metabarcode 1 (assumes format is like SILVA DB entries)

`db_other2`
: The reference database for other metabarcode 2 (assumes format is like SILVA DB entries)

Details

At this point, 'DADA2' function assignTaxonomy is used to assign taxonomy to the inferred ASVs.

Value

Taxonomic assignments of each unique ASV sequence

Examples

```
# Assign taxonomies to ASVs on by metabarcode
analysis_setup <- prepare_reads(
  data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  overwrite_existing = TRUE
)
cut_trim(
analysis_setup,
cutadapt_path="/usr/bin/cutadapt",
overwrite_existing = TRUE
)
make_asv_abund_matrix(
analysis_setup,
overwrite_existing = TRUE
)
assign_tax(
analysis_setup,
asv_abund_matrix,
retrieve_files=FALSE,
overwrite_existing = TRUE
)
```

convert_asv_matrix_to_objs *Filter ASV abundance matrix and convert to 'taxmap' and 'phyloseq' objects*

Description

Filter ASV abundance matrix and convert to 'taxmap' and 'phyloseq' objects

Usage

```
convert_asv_matrix_to_objs(
  analysis_setup,
  min_read_depth = 0,
  minimum_bootstrap = 0,
  save_outputs = FALSE
)
```

Arguments

analysis_setup	An object containing directory paths and data tables, produced by the prepare_reads function
min_read_depth	ASV filter parameter. If mean read depth of across all samples is less than this threshold, ASV will be filtered.
minimum_bootstrap	Set threshold for bootstrap support value for taxonomic assignments. Below designated minimum bootstrap threshold, taxonomic assignments will be set to N/A
save_outputs	Logical, indicating whether to save the resulting phyloseq and 'taxmap' objects. If TRUE, the objects will be saved; if FALSE, they will only be available in the global environment. Default is FALSE.

Value

ASV matrix converted to 'taxmap' object

Examples

```
# Convert final matrix to 'taxmap' and phyloseq objects for downstream analysis steps
analysis_setup <- prepare_reads(
data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  overwrite_existing = TRUE
)
cut_trim(
analysis_setup,
cutadapt_path="/usr/bin/cutadapt",
```

```
overwrite_existing = TRUE
)
make_asv_abund_matrix(
analysis_setup,
overwrite_existing = TRUE
)
assign_tax(
analysis_setup,
asv_abund_matrix,
retrieve_files=FALSE,
overwrite_existing=TRUE
)
objs<-convert_asv_matrix_to_objs(
analysis_setup
)
```

`cut_trim` *Main command to trim primers using 'Cutadapt' and core 'DADA2' functions*

Description

Main command to trim primers using 'Cutadapt' and core 'DADA2' functions

Usage

```
cut_trim(analysis_setup, cutadapt_path, overwrite_existing = FALSE)
```

Arguments

`analysis_setup`	An object containing directory paths and data tables, produced by the `prepare_reads` function
`cutadapt_path`	Path to the 'Cutadapt' program.
`overwrite_existing`	Logical, indicating whether to remove or overwrite existing files and directories from previous runs. Default is `FALSE`.

Details

If samples are comprised of two different metabarcodes (like ITS1 and rps10), reads will also be demultiplexed prior to 'DADA2'-specific read trimming steps.

Value

Trimmed reads, primer counts, quality plots, and ASV matrix.

Examples

```
# Remove remaining primers from raw reads, demultiplex pooled barcoded samples,
# and then trim reads based on specific 'DADA2' parameters
analysis_setup <- prepare_reads(
  data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  overwrite_existing = TRUE
)
cut_trim(
analysis_setup,
cutadapt_path="/usr/bin/cutadapt",
overwrite_existing = TRUE
)
```

`make_asv_abund_matrix` *Make an amplified sequence variant (ASV) abundance matrix for each of the input barcodes*

Description

Make an amplified sequence variant (ASV) abundance matrix for each of the input barcodes

Usage

```
make_asv_abund_matrix(analysis_setup, overwrite_existing = FALSE)
```

Arguments

`analysis_setup`	An object containing directory paths and data tables, produced by the `prepare_reads` function
`overwrite_existing`	Logical, indicating whether to overwrite existing results. Default is FALSE.

Details

The function processes data for each unique barcode separately, inferring ASVs, merging reads, and creating an ASV abundance matrix. To do this, the 'DADA2' core denoising alogrithm is used to infer ASVs.

Value

The ASV abundance matrix (`asv_abund_matrix`)

Examples

```
# The primary wrapper function for 'DADA2' ASV inference steps
analysis_setup <- prepare_reads(
  data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  overwrite_existing = TRUE
)
cut_trim(
analysis_setup,
cutadapt_path="/usr/bin/cutadapt",
overwrite_existing = TRUE
)
make_asv_abund_matrix(
analysis_setup,
overwrite_existing = TRUE
)
```

`prepare_reads` *Prepare reads for primer trimming using 'Cutadapt'*

Description

Prepare reads for primer trimming using 'Cutadapt'

Usage

```
prepare_reads(
  data_directory = "data",
  output_directory = tempdir(),
  tempdir_path = NULL,
  tempdir_id = "demulticoder_run",
  overwrite_existing = FALSE
)
```

Arguments

`data_directory`	Directory path where the user has placed raw FASTQ (forward and reverse reads), metadata.csv, and primerinfo_params.csv files. Default is "data".
`output_directory`	User-specified directory for outputs. Default is tempdir().
`tempdir_path`	Path to a temporary directory. If `NULL`, a temporary directory path will be identified using the `tempdir()` command.
`tempdir_id`	ID for temporary directories. The user can provide any helpful ID, whether it be a date or specific name for the run. Default is "demulticoder_run"
`overwrite_existing`	Logical, indicating whether to remove or overwrite existing files and directories from previous runs. Default is `FALSE`.

Value

A list containing data tables, including metadata, primer sequences to search for based on orientation, paths for trimming reads, and user-defined parameters for all subsequent steps.

Examples

```
# Pre-filter raw reads and parse metadata and primer_information to prepare
# for primer trimming and filter
analysis_setup <- prepare_reads(
  data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  overwrite_existing = TRUE
)
```

Index

assign_tax, 2

convert_asv_matrix_to_objs, 4
cut_trim, 5

make_asv_abund_matrix, 6

prepare_reads, 7