

Package ‘demulticoder’

March 17, 2025

Title An R Package for the Integrated Analysis of Multiplex Metabarcodes

Version 0.0.0.9000

Description The demulticoder package provides a workflow for analyzing multiplex metabarcoding data. It includes functions for filtering, trimming primers, and denoising data using Cutadapt and the dada2 package. The package can handle datasets with pooled amplicons and simple experimental designs.

License MIT

Encoding UTF-8

Depends R (>= 3.0.2)

Imports furrr,

purrr,
readr,
stringr,
tidyr,
dplyr,
ggplot2,
tibble,
utils,
knitr

Suggests BiocManager,

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

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

LazyData true

URL <https://grunwaldlab.github.io/demulticoder/>
BugReports <https://github.com/grunwaldlab/demulticoder/issues>
RoxygenNote 7.3.2
Config/Needs/website rmarkdown

R topics documented:

assign_tax	2
convert_asv_matrix_to_objs	4
cut_trim	5
make_asv_abund_matrix	6
prepare_reads	7
Index	9

assign_tax	<i>Assign taxonomy functions</i>
------------	----------------------------------

Description

Assign taxonomy functions

Usage

```
assign_tax(  
  analysis_setup,  
  asv_abund_matrix,  
  tryRC = FALSE,  
  verbose = FALSE,  
  multithread = FALSE,  
  retrieve_files = FALSE,  
  overwrite_existing = FALSE,  
  db_rps10 = "oomycetadb.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 ASV abundance matrix.
- tryRC Whether to try reverse complementing sequences during taxonomic assignment

verbose	Logical, indicating whether to display verbose output
multithread	Logical, indicating whether to use multithreading
retrieve_files	Specify TRUE/FALSE whether to copy files from the temp directory to the output directory
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 locus
db_its	The reference database for the ITS locus
db_16S	The reference database for the 16S locus
db_other1	The reference database for different locus 1 (assumes format is like SILVA DB entries)
db_other2	The reference database for a different locus 2 (assumes format is like SILVA DB entries)

Details

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

Value

Taxonomic assignments of each unique ASV sequence

Examples

```
# Assign taxonomies to ASVs on a per barcode basis
analysis_setup <- prepare_reads(
  data_directory = system.file("extdata", package = "demulticoder"),
  output_directory = tempdir(),
  tempdir_path = tempdir(),
  tempdir_id = "demulticoder_run_temp",
  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 `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 Threshold for bootstrap support value for taxonomic assignments. Below designated minimum bootstrap threshold, taxonomic assignments will be set to N/A

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(),
  tempdir_path = tempdir(),
  tempdir_id = "demulticoder_run_temp",
  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	<i>Main command to trim primers using Cutadapt and core DADA2 functions</i>
----------	---

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 barcodes (like ITS1 and rps10), reads will also be demultiplexed prior to DADA2 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(),
  tempdir_path = tempdir(),
  tempdir_id = "demulticoder_run_temp",
  overwrite_existing = TRUE
)
cut_trim(
  analysis_setup,
  cutadapt_path="/usr/bin/cutadapt",
  overwrite_existing = TRUE
)
```

make_asv_abund_matrix	<i>Make an amplified sequence variant (ASV) abundance matrix for each of the input barcodes</i>
-----------------------	---

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` `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 algorithm 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(),
  tempdir_path = tempdir(),
  tempdir_id = "demulticoder_run_temp",
  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 = "output",
  tempdir_path = NULL,
  tempdir_id = "demulticoder_run",
  overwrite_existing = FALSE
)
```

Arguments

<code>data_directory</code>	User-specified directory path where the user has placed raw FASTQ (forward and reverse reads), metadata.csv, and primerinfo_params.csv files. Default is "data".
<code>output_directory</code>	User-specified directory for outputs. Default is "output".
<code>tempdir_path</code>	Path to a temporary directory. If NULL, a temporary directory path will be identified using the <code>tempdir()</code> command.
<code>tempdir_id</code>	ID for temporary directories. Default is "demulticoder_run". The user can provide any helpful ID, whether it be a date or specific name for the run.

<code>overwrite_existing</code>	Logical, indicating whether to remove or overwrite existing files and directories from previous runs. Default is FALSE.
<code>multithread</code>	Logical, indicating whether to use multithreading for certain operations. 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(),
  tempdir_path = tempdir(),
  tempdir_id = "demulticoder_run_temp",
  overwrite_existing = TRUE
)
```


Index

`assign_tax`, [2](#)

`convert_asv_matrix_to_objs`, [4](#)

`cut_trim`, [5](#)

`make_asv_abund_matrix`, [6](#)

`prepare_reads`, [7](#)