Package 'cdmetapopR'

November 3, 2026

Title Utility Functions for 'CDMetaPOP'

Version 2.0.0

Date 2026-10-07

Description Provides a set of helper functions to launch, analyze, and visualize the output files of 'CDMetaPOP' <https://computationalecologylab.github.io/cdmetapop_web/>.

Imports ggplot2, R6, tidyr, tidyselect, dplyr, adegenet, gdistance, graph4lg, poppr, shiny, shinyBS, terra

License GPL (>= 2)

Encoding UTF-8

Suggests knitr, pkgload, rmarkdown, tidyverse, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Language en-US

RoxygenNote 8.0.0

NeedsCompilation no

Author Dominique Alò [aut, cre] (ORCID: <https://orcid.org/0000-0003-0579-4156>), Casey Day [aut], Allison Williams [aut], Marissa Roseman [aut]

Maintainer Dominique Alò <dominiquealo@gmail.com>

Repository CRAN

Date/Publication 2026-11-03 03:30:02 UTC

Contents

add_rows 2
age_structure_proportions 3
allele_frequencies_ind 4

as.data.frame.ClassVars . . . 5
cdmetapop_to_gene . . . 6
ClassVars . . . 6
create_cdmat . . . 9
DiseaseVars . . . 10
heterozygosity_ind . . . 12
launch_cdmetapop . . . 13
make_classvars . . . 15
make_diseasevars . . . 15
make_patchvars . . . 16
make_popvars . . . 16
make_runvars . . . 17
pairwise_fst_ind . . . 17
PatchVars . . . 18
PopVars . . . 23
read_cdmetapop . . . 29
RunVars . . . 30
summary_class . . . 32
summary_dataframe . . . 34
summary_disease . . . 35
summary_ind . . . 36
summary_patch_map . . . 38
summary_pop . . . 39
write_cdmetapop . . . 41

Index **43**

`add_rows` *Add one or more rows to a cdmetapopR input-file object*

Description

Generic function used to add rows to a `ClassVars()`, `PatchVars()`, `PopVars()`, or `RunVars()` object. Adds `n` new rows, each initialized as a copy of the current last row; edit the new row(s) afterward via the column active bindings (e.g. `myclassvars$maturation <- ...`). What a "row" represents depends on the class: an age class (`ClassVars()`), a patch (`PatchVars()`), a batch (`PopVars()`), or a run (`RunVars()`).

Usage

```
add_rows(x, n = 1)
```

Arguments

`x`	A cdmetapopR input-file object: a `ClassVars()`, `PatchVars()`, `PopVars()`, or `RunVars()` object.
`n`	Number of rows to add. Defaults to 1.

Details

A DiseaseVars() object holds exactly one row – CDMetaPOP reads a single disease model from DiseaseVars.csv and ignores any further rows – so calling this on one is an error rather than a no-op. Create another DiseaseVars() object to describe a second disease model.

Value

x, invisibly (modified in place, since x is an R6 object).

Examples

```
myclassvars <- ClassVars()
add_rows(myclassvars)
myclassvars$ages
```

---

age_structure_proportions

Determine the proportions of the age structure of the population

---

Description

This function summarizes the proportions of the age structure in the simulated population. It is calculated counting the values in the column 'age' of the chosen 'ind' generation.

Usage

```
age_structure_proportions(
  path = system.file("extdata", "Example_dat", package = "cdmetapopR"),
  runs = 1,
  gen = 49,
  species = 0
)
```

Arguments

path	path to the simulation folder
runs	integer referring to the number of montecarlo simulations output files to examine. It refers to the mcruns option in the RunVars CDMetaPOP input file. Right now it defaults to 1, which would be only one montecarlo run.
gen	Simulation run time (generation or year). Refers to runtime in the RunVars of CDMetaPOP input file. Currently defaults to 49.
species	If the simulation includes more species... The code for this needs to be adjusted

Value

a dataframe with a column for each montecarlo run (MC) with the proportion of the population for each given age (in rows) and the last column with the average value across all montecarlo runs at any given age.

Examples

```
mypath <- system.file("extdata", "Example_dat", package = "cdmetapopR")
age_structure_proportions(path = paste0(mypath, "/"), runs = 1, gen = 9)
```

`allele_frequencies_ind` *Plot Individual-Level Allele Frequencies*

Description

Calculates allele frequencies from `L#A#` genotype columns in CDMetaPOP `ind##.csv` or `ind##_Sample.csv` files.

Usage

```
allele_frequencies_ind(
  path,
  year = 0,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  file_type = "ind",
  patches = "all",
  loci = NULL,
  group_col = "PatchID",
  jitter = TRUE
)
```

Arguments

`path`	A dataframe, file path, vector of file paths, run directory, or top-level output directory containing individual files.
`year`	Integer. Year/generation to plot for one-year plots. Defaults to `0`.
`run`	Integer. Run index used when `path` is a directory. Defaults to `0`.
`batch`	Integer. Batch index used when `path` is a directory. Defaults to `0`.
`mc`	Integer. Monte Carlo index used when `path` is a directory. Defaults to `0`.
`species`	Integer. Species index used when `path` is a directory. Defaults to `0`.
`file_type`	Character. Which individual file type to read. Use `"ind"` for `ind##.csv` files or `"ind_Sample"` for `ind##_Sample.csv` files. Defaults to `"ind"`.
`patches`	Patch IDs to include. Use `"all"` to include all patches, a single patch ID such as `5`, or a vector/range such as `c(1, 3, 8)` or `1:20`. Defaults to `"all"`.
`loci`	Optional vector of loci to include, such as `c("L0", "L1")` or `c(0, 1)`. Defaults to all detected loci.

group_col	Column used for grouping frequencies. Defaults to "PatchID".
jitter	Logical. If TRUE, overlay jittered group-level points on the boxplots. Defaults to TRUE.

Value

A ggplot object with frequency data in plot$data.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")
allele_frequencies_ind(ex_dir, year = 9)
allele_frequencies_ind(ex_dir, year = 9, jitter = FALSE)
```

---

as.data.frame.ClassVars

View a cdmetapopR input-file object as a plain data frame

Description

Returns an ordinary (independent) data.frame copy of x's underlying table – e.g. for inspection with View(), or for any other base-R/ tidyverse workflow expecting a data frame. Editing the returned copy never affects x itself; edit x through its column active bindings (e.g. x$maturation <- ...) instead.

Usage

```
## S3 method for class 'ClassVars'
as.data.frame(x, ...)
```

Arguments

x	A cdmetapopR input-file object: a ClassVars(), PatchVars(), PopVars(), RunVars(), or DiseaseVars() object.
...	Currently unused; present for S3 consistency with base::as.data.frame().

Value

A data.frame.

Examples

```
myclassvars <- ClassVars()
as.data.frame(myclassvars)
```

cdmetapop_to_gene *Convert CDMetaPOP data to GENEPOP or GENALEX format*

Description

This function reads a CDMetaPOP CSV file, processes the genetic loci, and converts the data into GENEPOP or GENALEX formats.

Usage

```
cdmetapop_to_gene(path, format = "genepop", output_dir = tempdir())
```

Arguments

path	Character string; the file path to the CDMetaPOP CSV file.
format	A character string indicating the desired output format: "genepop" or "genalex". Genepop is default.
output_dir	Character string; the directory where output files will be written. Defaults to tempdir().

Value

The function writes the GENEPOP or GENALEX formatted data to the specified output directory.

Examples

```
path <- system.file("extdata", "Example_dat",
  "run0batch0mc0species0", "ind9.csv",
  package = "cdmetapopR")
cdmetapop_to_gene(path = path, format = "genepop")
cdmetapop_to_gene(path = path, format = "genalex")
```

ClassVars *Create a ClassVars object*

Description

Constructs an R6 object representing a CDMetaPOP ClassVars.csv input file: one row per age/size class, with one column per class-level parameter (mortality, migration, fecundity, etc.). Columns are edited as a whole after construction via $, e.g. myclassvars$age_mortality_out <- c(0, 0, 0.5). **To build the ClassVars object from an existing ClassVars csv file, use the 'path' argument or the read_cdmetapop() function.

Usage

```
ClassVars(
  ages = 0:5,
  body_size_mean = c(31, 53, 92, 123, 147, 184),
  body_size_std = c(0, 5, 0, 10, 10, 10),
  distribution = c(0.5, 0.25, 0.125, 0.0625, 0.03, 0.015),
  sex_ratio = rep(".50~.50", 6),
  age_mortality_out = rep(0, 6),
  age_mortality_out_stdev = rep(0, 6),
  age_mortality_back = c(0, 0, 0, 0, 0, 1),
  age_mortality_back_stdev = rep(0, 6),
  size_mortality_out = rep(0, 6),
  size_mortality_out_stdev = rep(0, 6),
  size_mortality_back = rep(0, 6),
  size_mortality_back_stdev = rep(0, 6),
  migration_out_prob = c(0, 0.1, 0.3, 0.5, 1, 1),
  migration_back_prob = rep(1, 6),
  straying_prob = rep(0.2, 6),
  dispersal_prob = rep(0.2, 6),
  maturation = c(0, 1, 1, 1, 1, 1),
  fecundity_ind = c(0, 5, 5, 10, 10, 10),
  fecundity_ind_stdev = rep(0, 6),
  fecundity_leslie = c(0, 2.5, 10, 11.7, 11.743, 15),
  fecundity_leslie_stdev = rep(0, 6),
  capture_out_prob = rep("N", 6),
  capture_back_prob = rep("N", 6),
  path = NULL
)
```

Arguments

`ages`	Integer vector of sequential age classes starting at 0 (e.g. `0:5`). Determines the number of rows; cannot be changed after construction (use `add_row()` or `add_rows()` instead).
`body_size_mean`, `body_size_std`	Mean and standard deviation of body size (units chosen by the user) at initialization, per age class.
`distribution`	Proportion of the population distributed to each age class at initialization. Each value in `[0, 1]`; does not need to sum to 1.
`sex_ratio`	Sex ratio per age class, as a `"~"`-separated string of 2-4 values (order: `FXX~MXY~MYY~FYY`) that must sum to 1, e.g. `".5~.5"`.
`age_mortality_out`, `age_mortality_back`	Age-specific mortality probability `[0, 1]` applied out of/back to the natal patch; or `"N"`; or `"~"`-separated per-sex values.
`age_mortality_out_stdev`, `age_mortality_back_stdev`	Standard deviation for the above; numeric, `"N"`, or `"~"`-separated.

`size_mortality_out`, `size_mortality_back`
: Size-specific mortality probability `[0, 1]`, `"N"`, or `"~"`-separated per-sex values.

`size_mortality_out_stdev`, `size_mortality_back_stdev`
: Standard deviation for the above; numeric, `"N"`, or `"~"`-separated.

`migration_out_prob`, `migration_back_prob`, `straying_prob`, `dispersal_prob`
: Movement probabilities `[0, 1]`, `"N"`, or `"~"`-separated per-sex values, multiplied by the corresponding values in PatchVars.

`maturation`
: Probability `[0, 1]` of being/becoming reproductively mature if sizeans from 'RunVars' = '"N"'; or `"N"`; or `"~"`-separated per-sex values.

`fecundity_ind`, `fecundity_ind_stdev`
: Mean and standard deviation of individual-based fecundity (litter/egg count) per age class if sizeans from 'RunVars' = '"N"'.

`fecundity_leslie`, `fecundity_leslie_stdev`
: Mean and standard deviation of Leslie-matrix fecundity per age class.

`capture_out_prob`, `capture_back_prob`
: Capture/detection probability `[0, 1]`, `"N"`, or `"~"`-separated per-sex values.

`path`
: Optional path to an existing ClassVars csv file. If supplied, the object is built from that file, with the age classes taken from its rows (so `ages` cannot also be given). Any column arguments supplied alongside `path` override the file's values, validated exactly as a later `$` assignment would be. With no overrides, this is equivalent to `read_cdmetapop(path, type = "ClassVars")`.

Details

Any column argument left as `NULL` defaults to the corresponding values in `example_files/classvars/ClassVarsAS1.csv`, matched to each requested age class (recycling the age-5 default, with a warning, for age classes beyond what that file defines).

Value

An R6 `ClassVars` object.

Examples

```
# Default 6-age-class object, matching ClassVarsAS1.csv:
myclassvars <- ClassVars()

# 3 age classes, defaults taken from ClassVarsAS1.csv's first 3 rows:
myclassvars <- ClassVars(ages = c(0, 1, 2))

# Edit a column in place:
myclassvars$age_mortality_out <- c(0, 0, 0.5)

# Add an age class (copies the last row; edit afterward):
myclassvars$add_row()
# or equivalently:
add_rows(myclassvars)
```

```
# Build from an existing ClassVars csv, overriding one column:
## Not run:
myclassvars <- ClassVars(path = "classvars/ClassVars_AS1.csv", maturation = 1)

## End(Not run)
```

`create_cdmat` *Create cost distance matrix for CDMetaPOP simulation*

Description

Takes input coordinates and produces a symmetrical cost distance matrix with options for Euclidean distance, equal distance, or least cost paths based on a provided resistance surface

Usage

```
create_cdmat(
  coords,
  method = c("euclidean", "equal", "lcp"),
  resistance = NULL
)
```

Arguments

`coords`	A set of spatial coordinates in the form of a 2-column data frame or matrix
`method`	Method for creating the cost distance matrix, must be "euclidean", "equal", or "lcp"
`resistance`	Raster object representing resistance surface for generating cost distance matrix

Details

Hello

Value

An NxN matrix of cost distances among patch locations

Examples

```
x <- c(1,2,3,4,5,6)
y <- c(1,2,3,4,5,6)
test <- data.frame(x=x, y=y)
create_cdmat(coords=test)

x <- runif(6, min=0, max=10)
y <- runif(6, min=0, max=10)
coords <- data.frame(x=x, y=y)
r <- terra::rast(nrows=10, ncols=10, xmin = 0, xmax = 10, ymin = 0, ymax = 10)
terra::values(r) <- 1
create_cdmat(coords=coords, method="lcp", resistance=r)
```

DiseaseVars *Create a DiseaseVars object*

Description

Constructs an R6 object representing a CDMetaPOP `DiseaseVars.csv` input file, referenced per patch by `PatchVars()`'s `disease_file` column. Columns may be edited after construction, e.g. `mydiseasevars$start_disease <- 10`. To build the object from an existing DiseaseVars file, use the `path` argument or the `read_cdmetapop()` function.

Usage

```
DiseaseVars(
  n_states = "3",
  initial_conditions = "0.9;0.1;0.0",
  transition_rates = "otherfiles/disease/TransitionMatrix_SIR.csv",
  susceptible_states = "0",
  infection_states = "1",
  death_states = "N",
  initial_conditions_offspring = "Susceptible",
  start_disease = "0",
  transmission_mode = "Direct",
  disease_resistant = "N",
  disease_tolerant = "N",
  path = NULL
)
```

Arguments

`n_states`	Number of disease compartments (a whole number, 1 or more). State 0 is always the initial susceptible state.
`initial_conditions`	Proportion of individuals initialized in each state at the start of the simulation, joined by `;`, one per state and summing to 1 (e.g. `"0.9;0.1;0.0"`).
`transition_rates`	The state-to-state transition matrix: either a square `matrix` or a path to a csv holding one. Rows are the state moved TO and columns the state moved FROM. A cell may be a single rate, or `"mu;sigma"` to redraw that rate from a normal distribution at each CDClimate time step.
`susceptible_states`	State index considered susceptible, or several joined by `;`.
`infection_states`	State index considered infectious, or several joined by `;`.
`death_states`	State index used for mortality or `"N"` for no death compartment.

`initial_conditions_offspring`
: How newborns' disease states are assigned: `"Susceptible"` (all born into state 0), or `"Vertical;<rate>"` for vertical transmission from infected mothers at that probability (e.g. `"Vertical;0.5"`).

`start_disease`
: Time step at which disease transmission begins (a non-negative whole number).

`transmission_mode`
: `"Direct"` (individual to individual) or `"Indirect"` (from a patch's environmental reservoir to individuals, which also enables the reservoir tracked by `PatchVars()`'s `env_res` column). Matched exactly, including case.

`disease_resistant`
: The transition(s) modified by the resistance locus, each written `"<from>_<to>"` with state indices and joined by `;` (e.g. `"0_1;3_1"`), or `"N"` for no resistance locus. Rates can vary by patch and with genotype, see `PatchVars()`.

`disease_tolerant`
: Same as `disease_resistant`, but for the tolerance locus.

`path`
: Optional path to an existing DiseaseVars csv file. If supplied, the object is built from that file. Any column arguments supplied alongside `path` override the file's values. With no overrides, this is equivalent to `read_cdmetapop(path, type = "DiseaseVars")`.

Details

Any column argument left unsupplied defaults to the basic 3-state SIR model from the user manual's disease quick-start.

The file holds only one row. `add_rows()` is not available for Disease parameters. Disease models may vary by patch by providing different `DiseaseVars` objects to different patches.

The disease module must also be turned on in `PopVars()`, by setting `implement_disease` to `"Out"`, `"Back"`, or `"Both"`, and each patch must reference a disease file in `PatchVars()`.

Delimiters: `;` is the only delimiter used in this file, separating the initial state proportions in `initial_conditions` and indicating which comparments correspond to which disease state. To vary disease parameters over time, use the `|` delimited in `PatchVars()`'s `disease_file` column.

States are numbered from 0, and state 0 is assumed to be the initial susceptible state. With `n_states = 3` the valid compartment numbers are 0, 1, and 2. Under `"Indirect"` transmission the environmental reservoir occupies one additional index (`n_states`), which must be included in `transition_rates` and can be the state named for `disease_resistant`/`disease_tolerant`.

Value

An R6 `DiseaseVars` object.

Examples

```
# The default 3-state SIR model:
mydiseasevars <- DiseaseVars()

# Start the disease dynamics at year 10 instead of year 0:
mydiseasevars$start_disease <- 10
```

```
# An SIR model supplying the transition matrix directly. Rows are the state
# moved TO, columns the state moved FROM: 0.5 is the S -> I rate and 0.2 the
# I -> R rate.
transitions <- matrix(0, nrow = 3, ncol = 3)
transitions[2, 1] <- 0.5
transitions[3, 2] <- 0.2
mydiseasevars <- DiseaseVars(transition_rates = transitions)

# Add vertical transmission from infected mothers:
mydiseasevars$initial_conditions_offspring <- "Vertical;0.3"

# Build from an existing DiseaseVars csv, overriding one column:
## Not run:
mydiseasevars <- DiseaseVars(
  path = "otherfiles/disease/DiseaseVars_SIR.csv",
  start_disease = 5
)

## End(Not run)
```

`heterozygosity_ind` *Plot Individual-Level Heterozygosity*

Description

Calculates observed and expected heterozygosity from `L#A#` genotype columns in CDMetaPOP `ind##.csv` files.

Usage

```
heterozygosity_ind(
  path,
  year = 0,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  file_type = "ind",
  patches = "all",
  loci = NULL,
  group_col = "PatchID",
  jitter = TRUE
)
```

Arguments

`path`	A dataframe, file path, vector of file paths, run directory, or top-level output directory containing individual files.

year	Integer. Year/generation to plot for one-year plots. Defaults to 0.
run	Integer. Run index used when path is a directory. Defaults to 0.
batch	Integer. Batch index used when path is a directory. Defaults to 0.
mc	Integer. Monte Carlo index used when path is a directory. Defaults to 0.
species	Integer. Species index used when path is a directory. Defaults to 0.
file_type	Character. Which individual file type to read. Use "ind" for ind##.csv files or "ind_Sample" for ind##_Sample.csv files. Defaults to "ind".
patches	Patch IDs to include. Use "all" to include all patches, a single patch ID such as 5, or a vector/range such as c(1, 3, 8) or 1:20. Defaults to "all".
loci	Optional vector of loci to include, such as c("L0", "L1") or c(0, 1). Defaults to all detected loci.
group_col	Column used for grouping frequencies. Defaults to "PatchID".
jitter	Logical. If TRUE, overlay jittered group-level points on the boxplots. Defaults to TRUE.

Value

A ggplot object with heterozygosity data in plot$data.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")
heterozygosity_ind(ex_dir, year = 9)
heterozygosity_ind(ex_dir, year = 9, jitter = FALSE)
```

---

launch_cdmetapop *Launch a CDMetaPOP simulation from R*

Description

Builds CDMetaPOP's input files from a RunVars() object (or uses an existing on-disk run directory) and launches the Python simulation.

When runvars is a RunVars() object, its full set of input files (RunVars -> PopVars -> PatchVars -> ClassVars, plus cost matrices and other referenced files) is written to a fresh, timestamped run directory (cdmetapop_run_<timestamp>/) created under base_dir, and CDMetaPOP is pointed at it. When runvars is instead a path to a directory containing a RunVars.csv, or directly to a RunVars csv file, that existing directory is used as-is and no input files are written.

CDMetaPOP writes its output into a timestamped subfolder of the run directory. Because CDMetaPOP runs can take a long time, the simulation launches in the background by default (wait = FALSE) so the R session is not blocked; set wait = TRUE to block until it finishes. Either way, read results afterward by pointing the summary_* functions (e.g. summary_pop()) at the returned run directory.

Usage

```
launch_cdmetapop(
  runvars,
  pythonFilepath = "python",
  CDMetaPOPFilepath = "CDMetaPOP.py",
  base_dir = getwd(),
  output_prefix = "output_",
  wait = FALSE
)
```

Arguments

runvars	A RunVars() object (its inputs are written to a fresh run directory), or a character path to an existing run directory or RunVars.csv file.
pythonFilepath	Location of the Python executable, or just "python" if it is already on the PATH.
CDMetaPOPFilepath	Location of CDMetaPOP's CDMetaPOP.py, or just "CDMetaPOP.py" if it is in the working directory.
base_dir	Parent directory under which the fresh run directory is created when runvars is a RunVars() object. Defaults to the working directory. Ignored when runvars is a path to an existing run.
output_prefix	Name prefix for CDMetaPOP's output subfolder; CDMetaPOP appends a unix timestamp to it. Defaults to "output_".
wait	If FALSE (the default), launch the simulation in the background and return immediately – suitable for long runs. If TRUE, block until CDMetaPOP finishes before returning. On macOS and Linux a background run opens no console window, so CDMetaPOP's output may not reach the R console; check its CDmetaPOP0.log file in the output folder for progress and errors (a warning on launch says the same).

Value

The output run directory, to be used as an argument for output summary functions, (e.g. summary_pop()) once the run has completed.

Examples

```
## Not run:
# Build inputs from a RunVars object and launch in the background:
myrun <- RunVars()
run_dir <- launch_cdmetapop(
  myrun,
  pythonFilepath = "C:/Users/User1/anaconda3/python.exe",
  CDMetaPOPFilepath = "C:/Users/User1/CDMetaPOP/src/CDMetaPOP.py"
)
# ... once the run finishes, read results from the same directory:
summary_pop(run_dir, type = "N_initial")
```

```
# Or launch an existing on-disk run directory, blocking until it finishes:
launch_cdmetapop("C:/Users/User1/CDMetaPOP/example_files", wait = TRUE)

## End(Not run)
```

`make_classvars` *Build ClassVars File from Template*

Description

This function loads a provided ClassVars template, allows the user to edit selected areas, and saves the modified version as a new file.

Usage

```
make_classvars(output_file = "my_new_classvars.csv")
```

Arguments

`output_file`	The name of the output file. Defaults to 'my_new_classvars.csv'.

Value

A Shiny app instance.

`make_diseasevars` *Build DiseaseVars File from Template*

Description

This function loads a provided DiseaseVars template, allows the user to edit selected areas, and saves the modified version as a new file.

Usage

```
make_diseasevars(output_file = "my_new_diseasevars.csv")
```

Arguments

`output_file`	The name of the output file. Defaults to 'my_new_diseasevars.csv'.

Value

A Shiny app instance.

make_patchvars *Build PatchVars File from Template*

Description

This function loads a provided PatchVars template, allows the user to edit selected areas, and saves the modified version as a new file.

Usage

```
make_patchvars(output_file = "my_new_patchvars.csv")
```

Arguments

output_file	The name of the output file. Defaults to 'my_new_patchvars.csv'.

Value

A Shiny app instance.

make_popvars *Build PopVars File from Template*

Description

This function loads a provided PopVars template, allows the user to edit selected areas, and saves the modified version as a new file.

Usage

```
make_popvars(output_file = "my_new_popvars.csv")
```

Arguments

output_file	The name of the output file. Defaults to 'my_new_popvars.csv'.

Value

A Shiny app instance.

make_runvars *Build RunVars File from Template*

Description

This function loads a provided runVars template, allows the user to edit selected areas, and saves the modified version as a new file.

Usage

```
make_runvars(output_file = "my_new_runvars.csv")
```

Arguments

output_file	The name of the output file. Defaults to 'my_new_runvars.csv'.

Value

A Shiny app instance.

pairwise_fst_ind *Plot Pairwise FST Between Individual Groups*

Description

Calculates pairwise FST between groups, usually patches, from L#A# genotype columns in CDMetaPOP ind##.csv files. FST is calculated per locus and then averaged across loci for each pair.

Usage

```
pairwise_fst_ind(
  path,
  year = 0,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  file_type = "ind",
  patches = "all",
  loci = NULL,
  group_col = "PatchID"
)
```

Arguments

path	A dataframe, file path, vector of file paths, run directory, or top-level output directory containing individual files.
year	Integer. Year/generation to plot for one-year plots. Defaults to 0.
run	Integer. Run index used when path is a directory. Defaults to 0.
batch	Integer. Batch index used when path is a directory. Defaults to 0.
mc	Integer. Monte Carlo index used when path is a directory. Defaults to 0.
species	Integer. Species index used when path is a directory. Defaults to 0.
file_type	Character. Which individual file type to read. Use "ind" for ind##.csv files or "ind_Sample" for ind##_Sample.csv files. Defaults to "ind".
patches	Patch IDs to include. Use "all" to include all patches, a single patch ID such as 5, or a vector/range such as c(1, 3, 8) or 1:20. Defaults to "all".
loci	Optional vector of loci to include, such as c("L0", "L1") or c(0, 1). Defaults to all detected loci.
group_col	Column used for grouping frequencies. Defaults to "PatchID".

Value

A ggplot heatmap with pairwise FST data in plot$data.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")
pairwise_fst_ind(ex_dir, year = 9)
```

PatchVars *Create a PatchVars object*

Description

Constructs an R6 object representing a CDMetaPOP PatchVars.csv input file: one row per patch, with one column per patch-level parameter (carrying capacity, mortality, migration, growth, fitness, etc.). Columns are edited as a whole after construction via $, e.g. mypatchvars$k <- c(300, 300, 500). **To build the PatchVars object from an existing PatchVars csv file, use the 'path' argument or the read_cdmetapop() function.

Usage

```
PatchVars(
  patch_id = 1:7,
 x = c(2540470.832, 2536859.926, 2532969.44, 2539041.489, 2535011.582, 2545325.475,
    2527429.642),
 y = c(712452.2021, 708059.4624, 705413.5711, 728416.6747, 720257.8531, 726552.0418,
    708511.8011),
```

```
subpatch_no = c("1", "1", "2", "2", "3", "3", "3"),
k = rep(300, 7),
k_stdev = rep(0, 7),
n0 = rep(150, 7),
natal_grounds = rep(1, 7),
migration_grounds = rep(1, 7),
genes_initialize = c("genes/allelefrequencyA.csv", rep("random", 6)),
class_vars = rep("classvars/ClassVars_AS1.csv", 7),
mortality_out = rep(0, 7),
mortality_out_stdev = rep(0, 7),
mortality_back = rep(0, 7),
mortality_back_stdev = rep(0, 7),
mortality_eggs = rep(0, 7),
mortality_eggs_stdev = rep(0, 7),
migration_out_prob = rep(1, 7),
set_migration = rep("Y", 7),
migration_back_prob = rep(1, 7),
straying_prob = rep(1, 7),
dispersal_prob = rep(1, 7),
growth_temp_out = rep(1, 7),
growth_temp_out_stdev = c(0, 0.2, 0.1, 1, 0.4, 0.2, 0),
grow_days_out = c(155, 150, 150, 150, 150, 150, 150),
grow_days_out_stdev = c(0, 20, 20, 20, 20, 20, 20),
growth_temp_back = c(14, 18, 22, 8, 8, 8, 8),
growth_temp_back_stdev = rep(0, 7),
grow_days_back = c(210, 150, 150, 150, 150, 150, 150),
grow_days_back_stdev = c(0, 20, 20, 20, 20, 20, 20),
capture_prob_out = rep("N", 7),
capture_prob_back = rep("N", 7),
habitat_out = rep(1, 7),
habitat_back = rep(1, 7),
fitness_AA = rep(0, 7),
fitness_Aa = rep(0, 7),
fitness_aa = rep(0, 7),
fitness_BB = rep(0, 7),
fitness_Bb = rep(0, 7),
fitness_bb = rep(0, 7),
fitness_AABB = rep(0, 7),
fitness_AaBB = rep(0, 7),
fitness_aaBB = rep(0, 7),
fitness_AABb = rep(0, 7),
fitness_AaBb = rep(0, 7),
fitness_aaBb = rep(0, 7),
fitness_AAbb = rep(0, 7),
fitness_Aabb = rep(0, 7),
fitness_aabb = rep(0, 7),
comp_coef = rep("0.5;0.1", 7),
disease_file = NULL,
```

```
  env_res = NULL,
  resistant_CC = NULL,
  resistant_Cc = NULL,
  resistant_cc = NULL,
  tolerant_DD = NULL,
  tolerant_Dd = NULL,
  tolerant_dd = NULL,
  resolve_class_vars = TRUE,
  resolve_disease_file = TRUE,
  path = NULL
)
```

Arguments

`patch_id`	Integer vector of sequential patches starting at 1 (e.g. `1:7`). Determines the number of rows; cannot be changed after construction (use `add_row()` or `add_rows()` instead).
`x, y`	Coordinate locations for each patch. The only columns besides `patch_id`/`subpatch_no` that do NOT support the `|` temporal-change mechanism described in Details.
`subpatch_no`	Identifier (numeric or character) tagging individuals to a particular region; reported in output files. Does not support `|` (a one-time identifier, not a value that changes over a run).
`k, k_stdev`	Carrying capacity and its annual standard deviation. `0` means individuals cannot move into the patch. A new value with SD is drawn each year of the simulation.
`n0`	Number of individuals to initialize the patch at year 0.
`natal_grounds, migration_grounds`	`0`/`1` flags for whether individuals may occupy the patch at natal grounds ("back") or migration/overwintering grounds ("out") respectively.
`genes_initialize`	Genotype initialization. Per patch, a value is `"random"`, `"random_var"`, or an allele-frequency file path. The `|` temporal-change mechanism applies (see Details); within each `|` group, multiple allele-frequency files may be `;`-separated, but the `"random"`/`"random_var"` keywords must each appear alone in a group (never `;`-joined). E.g. `"f1.csv;f2.csv|random"` is valid; `"random;f1.csv"` is not.
`class_vars`	A `ClassVars()` object, filepath, or object-name string (`;`-separated for multiple and '"|"' -separated for temporal variation). See Details for more information.
`mortality_out, mortality_back, mortality_eggs`	Density-independent mortality `[0, 1]`, `"N"` (no patch-level mortality), or `"E"` (eradication override).
`mortality_out_stdev, mortality_back_stdev, mortality_eggs_stdev`	Standard deviation for the above; numeric, `"N"`, or `"E"`.
`migration_out_prob`	Emigration probability `[0, 1]`, multiplied by the corresponding value in ClassVars.
`set_migration`	`"Y"` or `"N"`: whether a migrant individual stays a migrant with probability 1.

migration_back_prob, straying_prob, dispersal_prob
: Movement probabilities [0, 1], multiplied by the corresponding values in ClassVars.

growth_temp_out, growth_temp_back
: Temperature values influencing body size growth; numeric or "N" (turn off).

growth_temp_out_stdev, growth_temp_back_stdev
: Standard deviation for the above; numeric.

grow_days_out, grow_days_back
: Growing days [0, 365], or "N" (ignored).

grow_days_out_stdev, grow_days_back_stdev
: Standard deviation for the above; numeric.

capture_prob_out, capture_prob_back
: Capture/detection probability [0, 1], or "N" (ignore).

habitat_out, habitat_back
: Habitat quality values for the plasticity module.

fitness_AA, fitness_Aa, fitness_aa, fitness_BB, fitness_Bb, fitness_bb, fitness_AABB, fitness_AaBB, fitness_aaBB, fitness_AABb, fitness_AaBb, fitness_aaBb, fitness_AAbb, fitness_Aabb, fitness_aabb
: Genotype-specific fitness/selection values; format depends on PopVars$cdevolveans (not yet implemented), so validation here is permissive.

comp_coef
: Lotka-Volterra competition coefficient(s) for multispecies applications, ;-separated if more than 2 species.

disease_file
: A DiseaseVars() object, object-name/path string, or several joined with | (successive time steps), for each patch. Required – along with the seven other disease arguments – when PopVars()'s implement_disease is not "N". Several patches may share one DiseaseVars object. See Details.

env_res
: Starting pathogen load in each patch's environmental reservoir (a non-negative number). Only used when the referenced DiseaseVars() sets transmission_mode = "Indirect", but CDMetaPOP requires a number either way; use 0 when transmission is direct.

resistant_CC, resistant_Cc, resistant_cc
: Transition rate for each genotype at the disease-resistance locus, overriding the matching cell of the DiseaseVars() transition matrix. Give one rate per transition named in that file's disease_resistant field, ;-separated in the same order (e.g. "0.2;0.9" for "0_1;3_1").

tolerant_DD, tolerant_Dd, tolerant_dd
: As the resistant_* arguments, for the disease-tolerance locus and its disease_tolerant field.

resolve_class_vars
: If TRUE, strings entered for class_vars will be checked against the global environment for ClassVars objects of the same name (primarily helpful for temporal changes e.g., "classvars1|classvars2"). If FALSE, class_vars entries must be ClassVars objects or they will be treated as filepaths.

resolve_disease_file
: As resolve_class_vars, for disease_file strings and DiseaseVars() objects.

path — Optional path to an existing PatchVars csv file. If supplied, the object is built from that file, with the patches taken from its rows (so patch_id cannot also be given). Any column arguments supplied alongside path override the file's values, validated exactly as a later $ assignment would be. Values read from the file are always treated as literal file paths, so resolve_class_vars applies only when path is NULL. With no overrides, this is equivalent to read_cdmetapop(path, type = "PatchVars").

Details

Any column argument left as NULL defaults to the corresponding values in example_files/patchvars/PatchVarsS1.csv, matched to each requested patch (recycling the patch-7 default, with a warning, for patches beyond what that file defines). The eight disease arguments are the exception: they have no default and are left out of the file entirely unless set (see Details).

Disease columns. disease_file, env_res, and the six resistant_*/tolerant_* rates are only part of a PatchVars file when the disease module is on. CDMetaPOP requires all eight together: it expects exactly 50 columns when PopVars()'s implement_disease is "N" and exactly 58 when it is "Out", "Back", or "Both", and rejects either shape in the other mode. So leaving them unset – the default – produces the 50-column non-disease file, setting all eight produces the 58-column one, and setting only some warns, since no file with a partial set can run. They are absent from PatchVarsS1.csv, which is why they alone have no default values. Disease states, transition rates, and compartments are defined in the referenced DiseaseVars() file, not here; these columns only say which model each patch follows and give the per-patch, per-genotype values.

Temporal changes via "|": CDMetaPOP lets most PatchVars columns specify a value that changes based on values in the RunVars object (cdclimate module), by giving multiple values separated by | in one cell, e.g. mypatchvars$mortality_out <- "0|0.2|0.5" (mortality is 0 at first, then 0.2, then 0.5, switching at the timesteps configured in RunVars). This applies to every column except x, y, patch_id, and subpatch_no. Because this is a single delimited *string*, not a vector, it must always be entered as one quoted character value (e.g. "0|0.2|0.5"), even for columns that are otherwise numeric – assigning a plain numeric vector like c(0, 0.2, 0.5) would instead be interpreted as three different patches' worth of values, not one patch's value over time. Each |-separated segment is validated independently against the column's usual rule (numeric bounds, "N"/"E", etc.).

class_vars and genes_initialize may also be entered as strings, for consistency with every other column's "|" mechanism above, even though they conceptually reference other R objects/files rather than plain values. For class_vars specifically: each |-separated segment of the string is checked against the name of every ClassVars object that exists in the global environment; if one matches, that *object* is used, otherwise the segment is treated as a literal file path, same as genes_initialize always is. For example, mypatchvars$class_vars <- "cv_year0|cv_year5" will use the actual cv_year0/cv_year5 objects if they exist as that-named variables in your global environment, or otherwise treat those strings as paths to existing ClassVars csv files. Building your ClassVars objects inside a function rather than at the global level will not be found this way (only the global environment is searched, deliberately – see .pv_resolve_class_vars_segment()); in that case, assign the actual object(s) directly instead (e.g. mypatchvars$class_vars <- cv_year0, or list(cv_year0, cv_year5) for multiple), without quoting.

A literal path string could, in principle, coincidentally match the name of an unrelated ClassVars object in your global environment and get resolved to that object instead of being treated as text.

As a precaution, it's good practice to keep ClassVars/allele-frequency csv files in their own subdirectories (e.g. classvars/, genes/) regardless of how you plan to reference them – this keeps the two namespaces (file paths vs. object names) from ever plausibly colliding.

Value

An R6 PatchVars object.

Examples

```
# Default 7-patch object, matching PatchVarsS1.csv:
mypatchvars <- PatchVars()

# 3 patches, defaults taken from PatchVarsS1.csv's first 3 rows:
mypatchvars <- PatchVars(patch_id = 1:3)

# 3 patches with one column defined and other columns as defaults:
mypatchvars <- PatchVars(patch_id = 1:3, natal_grounds = c(1,1,0))

# Edit a column in place:
mypatchvars$k <- c(300, 300, 500)

# Add a patch (copies the last row; edit afterward):
mypatchvars$add_row()
# or equivalently:
add_rows(mypatchvars)

# Build from an existing PatchVars csv, overriding one column:
## Not run:
mypatchvars <- PatchVars(path = "patchvars/PatchVarsS1.csv", k = 500)

## End(Not run)
```

PopVars *Create a PopVars object*

Description

Constructs an R6 object representing a CDMetaPOP PopVars.csv input file. Unlike ClassVars() (rows = age classes) and PatchVars() (rows = patches), each row of a PopVars object is its own independent "batch" simulation set. CDMetaPOP runs each RunVars row (a "run") x each PopVars row (a "batch") as a separate simulation (see the user manual's run-parameters section). Columns may be edited as a whole via $, e.g. mypopvars$matemoveno <- c(6, 6, 4, 6). **To build the PopVars object from an existing PopVars csv file, use the 'path' argument or the read_cdmetapop() function.

Usage

```
PopVars(
  n_batches = 4,
  xyfilename = c("patchvars/PatchVars.csv", "patchvars/PatchVars_anadromy.csv",
    "patchvars/PatchVars.csv", "patchvars/PatchVars.csv"),
 mate_cdmat = c("cdmats/Patch7_CdmatrixS1.csv", "cdmats/Patch7_CdmatrixS1.csv",
    "cdmats/Patch7_CdmatrixS1.csv", "cdmats/Patch7_CdmatrixS1.csv"),
  matemoveno = c("6", "6", "4", "6"),
  matemoveparA = c("1.346", "1.346", "1.346", "1.346"),
  matemoveparB = c("202.4846", "202.4846", "202.4846", "202.4846"),
  matemoveparC = c("6", "6", "6", "6"),
  matemovethresh = c("6000", "6000", "6000", "6000"),
 migrateout_cdmat = c("cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv"),
  migratemoveOutno = c("4", "4", "9", "9"),
  migratemoveOutparA = c("0", "0", "0", "0"),
  migratemoveOutparB = c("0", "0", "0", "0"),
  migratemoveOutparC = c("0", "0", "0", "0"),
  migratemoveOutthresh = c("max~max", "max~max", "max~max", "max~max"),
 migrateback_cdmat = c("cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv",
    "cdmats/Patch7_Probmatrix_onewayBarriersXRiverineS1.csv"),
  migratemoveBackno = c("4", "4", "9", "9"),
  migratemoveBackparA = c("0", "0", "0", "0"),
  migratemoveBackparB = c("0", "0", "0", "0"),
  migratemoveBackparC = c("0", "0", "0", "0"),
  migratemoveBackthresh = c("max~max", "max~max", "max~max", "max~max"),
 stray_cdmat = c("cdmats/Patch7_CdmatrixS1.csv", "cdmats/Patch7_CdmatrixS1.csv",
    "cdmats/Patch7_CdmatrixS1.csv", "cdmats/Patch7_CdmatrixS1.csv"),
  StrayBackno = c("1", "1", "4", "2"),
  StrayBackparA = c("0.01", "0.01", "0.01", "0.01"),
  StrayBackparB = c("0.01", "0.01", "0.01", "0.01"),
  StrayBackparC = c("0", "0", "0", "0"),
  StrayBackthresh = c("max", "max", "max", "max"),
  disperseLocal_cdmat = c("cdmats/Patch7_Probmatrix_Dispersal.csv",
    "cdmats/Patch7_Probmatrix_Dispersal.csv", "cdmats/Patch7_CdmatrixS1.csv",
    "cdmats/Patch7_CdmatrixS1.csv"),
  disperseLocalno = c("9", "9", "2", "2"),
  disperseLocalparA = c("0.01", "0.01", "0.01", "0.01"),
  disperseLocalparB = c("0.01", "0.01", "0.01", "0.01"),
  disperseLocalparC = c("0", "0", "0", "0"),
  disperseLocalthresh = c("max", "max", "max", "max"),
  HomeAttempt = c("mortality", "mortality", "mortality", "mortality"),
  sex_chromo = c("2", "2", "2", "2"),
  sexans = c("Y", "Y", "Y", "H"),
```

```
 selfans = c("N", "N", "N", "0.5"),
 Freplace = c("Y", "Y", "Y", "Y"),
 Mreplace = c("Y", "Y", "Y", "Y"),
 AssortativeMate_Model = c("1", "1", "1", "1"),
 AssortativeMate_Factor = c("1", "1", "1", "1"),
 mature_default = c("age3|age6|age3", "age3", "size300", "N"),
 mature_eqn_slope = c("0.13~0.06|0.13~0.06|0.13~0.06", "0.0539", "0.13~0.06",
   "0.13~0.06"),
mature_eqn_int = c("-20.28~-8.09|-20.28~-8.09|-20.28~-8.09", "-6.313", "-20.28~-8.09",
   "-20.28~-8.09"),
 offno = c("2", "2", "2", "2"),
 offans_InheritClassVars = c("random", "random", "random", "random"),
 equalClutchSize = c("N", "N", "N", "N"),
 Egg_Freq_Mean = c("1", "1", "1", "0.5"),
 Egg_Freq_StDev = c("1", "1", "0", "0.1"),
 Egg_Mean_ans = c("linear", "linear", "exp", "exp"),
 Egg_Mean_par1 = c("-445", "-445", "126.07", "126.07"),
 Egg_Mean_par2 = c("3.78", "3.78", "0.0061", "0.0061"),
 Egg_Mortality = c("N", "0.62", "0.62", "0"),
 Egg_Mortality_StDev = c("0", "0", "0.18", "0"),
 Egg_FemaleProb = c("0.5", "0.5", "0.5", "0.5"),
 startGenes = c("0", "0", "0", "0"),
 loci = c("2", "2", "2", "2"),
 alleles = c("2", "2", "2", "2"),
 muterate = c("0", "0", "0", "0"),
 mutationtype = c("random", "random", "random", "random"),
 mtdna = c("N", "N", "N", "N"),
 cdevolveans = c("N", "N", "N", "N"),
 startSelection = c("0", "0", "0", "0"),
 implementSelection = c("Out:Back", "Out:Back", "Eggs", "Back"),
 betaFile_selection = c("N", "N", "N", "N"),
 plasticgeneans = c("N", "N", "N", "N"),
 plasticSignalResponse = c("0", "0", "0", "0"),
 plasticBehavioralResponse = c("0", "0", "0", "0"),
 startPlasticgene = c("0", "0", "0", "0"),
 implementPlasticgene = c("Back:0", "Back:0", "Back", "Back"),
growth_option = c("temperature", "temperature", "temperature", "temperature"),
 growth_Loo = c("400", "400", "250", "250"),
 growth_R0 = c("0.47", "0.47", "0.57", "0.57"),
 growth_temp_max = c("10.5", "10.5", "12", "12"),
 growth_temp_CV = c("0.33", "0.33", "0.25", "0.25"),
 growth_temp_t0 = c("-0.075", "-0.075", "-0.196", "-0.196"),
 popmodel = c("packing", "anadromy", "logistic_back", "N"),
 popmodel_par1 = c("-0.6821", "-0.6821", "-0.6821", "-0.6821"),
 correlation_matrix = c("N", "N", "N", "N"),
 subpopmort_file = c("N", "N", "N", "N"),
 egg_delay = c("0", "0", "0", "0"),
 egg_add = c("mating", "mating", "mating", "mating"),
```

```
  implement_disease = c("N", "N", "N", "N"),
  resolve_xyfilename = TRUE,
  path = NULL
)
```

Arguments

`n_batches`	Number of independent batches. Defaults to 4, matching `PopVars.csv`. Can be changed after construction using `add_row()` or `add_rows()`.
`xyfilename`	A `PatchVars()` object, object name, or path string, for each batch.
`mate_cdmat`, `migrateout_cdmat`, `migrateback_cdmat`, `stray_cdmat`, `disperseLocal_cdmat`	A cost-distance/probability `matrix` object (single variable only) or a file path (`|`/`~`-delimited for temporal/per-sex). See Details.
`matemoveno`, `migratemoveOutno`, `migratemoveBackno`, `StrayBackno`, `disperseLocalno`	Movement function code, `1`-`11` (see manual for the function each code selects).
`matemoveparA`, `matemoveparB`, `matemoveparC`, `migratemoveOutparA`, `migratemoveOutparB`, `migratemoveOutparC`, `migratemoveBackparA`, `migratemoveBackparB`, `migratemoveBackparC`, `StrayBackparA`, `StrayBackparB`, `StrayBackparC`, `disperseLocalparA`, `disperseLocalparB`, `disperseLocalparC`	Movement function parameters A/B/C (meaning depends on the corresponding `*no` function code).
`matemovethresh`, `migratemoveOutthresh`, `migratemoveBackthresh`, `StrayBackthresh`, `disperseLocalthresh`	Movement threshold: `"max"`, `"<percent>max"` (e.g. `"10max"`), or a cost-distance value.
`HomeAttempt`	How a migrant that cannot return home is handled: `"mortality"`, `"stray_emiPop"`, or `"stray_natalPop"`.
`sex_chromo`	Number of sex chromosome combinations: `2`, `3`, or `4`.
`sexans`	Reproduction mode: `"Y"` (sexual), `"N"` (asexual), or `"H"` (hermaphroditic).
`selfans`	Selfing: `"Y"`, `"N"`, or a `[0, 1]` probability (for `sexans = "H"`).
`Freplace`, `Mreplace`	Mating-with-replacement flags, `"Y"`/`"N"`.
`AssortativeMate_Model`	Assortative mating model: `"1"`, `"2"`, `"3a"`, `"3b"`, `"4"`, or `"5"`.
`AssortativeMate_Factor`	Assortative mating factor, `[1, 1000000]`.
`mature_default`	Default maturation trigger, e.g. `"age6"`, `"size200"`, or `"N"`.
`mature_eqn_slope`, `mature_eqn_int`	Size-based maturation probability function parameters.
`offno`	Offspring-count draw choice: `1` (uniform), `2` (Poisson), `3` (constant), or `4` (normal).

`offans_InheritClassVars`
: How offspring inherit `ClassVars` when multiple are specified: `"random"`, `"Hindex"`, or `"mother"`.

`equalClutchSize`
: `"Y"`/`"N"`: equal clutch size per mate pair.

`Egg_Freq_Mean`, `Egg_Freq_StDev`
: Egg-laying frequency mean/standard deviation.

`Egg_Mean_ans`
: Size-based fecundity function: `"exp"`, `"linear"`, or `"pow"`.

`Egg_Mean_par1`, `Egg_Mean_par2`
: Parameters for the above function.

`Egg_Mortality`, `Egg_Mortality_StDev`
: Egg mortality `[0, 1]` (or `"N"`) and its standard deviation.

`Egg_FemaleProb`
: Probability an egg is female, `[0, 1]`, or `"WrightFisher"`.

`startGenes`
: Time unit genetic exchange begins.

`loci`
: Number of loci (>= 2)

`alleles`
: number of starting alleles per locus (>= 2), or `:`-separated per-locus counts (e.g. `"2:5:3"`).

`muterate`
: Allele mutation rate, `[0, 1]`.

`mutationtype`
: Mutation model: `"random"`, `"forward"`, `"backward"`, `"forwardbackward"`, or `"forwardAbackwardBrandomN"`.

`mtdna`
: `"Y"`/`"N"`: track a maternal (mtDNA) marker locus.

`cdevolveans`
: Selection model/mechanism (open-ended; see manual for the full grammar, e.g. `"N"`, `"1"`, `"M"`, `"Hindex_Gauss_..."`).

`startSelection`
: Time unit selection begins.

`implementSelection`
: Timing of selection e.g., 'Out', 'Back', 'Eggs', 'Out:Back:Eggs', etc.

`betaFile_selection`
: Polygenic-selection beta-value file path, or `"N"`.

`plasticgeneans`
: Phenotypic plasticity signal/response spec (open- ended; see manual), or `"N"`.

`plasticSignalResponse`, `plasticBehavioralResponse`
: Plasticity signal trigger value and behavioral response threshold.

`startPlasticgene`
: Time unit the plastic process begins.

`implementPlasticgene`
: Timing of plastic gene implementation ('Out', 'Back', or 'Out:Back')

`growth_option`
: Growth function: `"N"`, `"known"`, `"vonB"`, `"temperature"`, `"temperature_hindex"`, or `"bioenergetics"`.

`growth_Loo`
: Von Bertalanffy L-infinity, or two `;`-separated genotype-linked values (`"Loo_1;Loo_2"`).

`growth_R0`, `growth_temp_max`, `growth_temp_CV`, `growth_temp_t0`
: Remaining growth function parameters.

`popmodel`
: Population growth model: `"N"`(exponential), `"logistic_out"`, `"logistic_back"`, `"packing"`, or `"anadromy"`.

popmodel_par1 Parameter for the "packing" models.

correlation_matrix
Patch-level correlation matrix: a matrix (single value), a file path, or "N". A matrix field – see Details.

subpopmort_file
Sub-population dispersal mortality matrix: a matrix (single value), a file path, or "N". A matrix field – see Details.

egg_delay Integer >= 0: years between mating and gestation/emergence.

egg_add "mating" or "nonmating" (case-insensitive).

implement_disease
Disease module timing: "N", "Back", "Out", or "Both".

resolve_xyfilename
If TRUE, strings entered for xyfilename will be checked against the global environment for PatchVars objects of the same name (primarily helpful for temporal changes e.g., "patchvars1|patchvars2"). If FALSE, xyfilename entries must be PatchVars objects or they will be treated as filepaths.

path Optional path to an existing PopVars csv file. If supplied, the object is built from that file, with the number of batches taken from its row count (so n_batches cannot also be given). Any column arguments supplied alongside path override the file's values, validated exactly as a later $ assignment would be. Values read from the file are always treated as literal file paths, so resolve_xyfilename applies only when path is NULL. With no overrides, this is equivalent to read_cdmetapop(path, type = "PopVars").

Details

Any column argument left unsupplied defaults to the corresponding values in example_files/popvars/PopVars.csv, matched to each requested batch (recycling the row-4 default, with a warning, for rows beyond what that file defines).

CDMetaPOP reads PopVars.csv headers as dictionary keys, so every column name below is used verbatim as both the data frame's column name and this function's argument name.

Delimiters: most columns support CDMetaPOP's | (temporal change) and/or ~ (per-sex) mechanisms. A few fields use other manual-documented delimiters for a specific purpose: alleles and implementSelection/implementPlasticgene use : (per-locus allele counts; ':'-joined timing keywords, respectively), and growth_Loo uses ; for its two genotype-linked values (Loo_1;Loo_2) – see .popv_rules' doc comment in class_popvars.R for the full per-field delimiter design and known open questions.

xyfilename (PatchVars.csv file pointer) accepts a PatchVars() object directly, a character path, or the name of a PatchVars object in .GlobalEnv (see PatchVars()'s Details for the identical class_vars name-resolution mechanism and its caveats).

Matrix fields (mate_cdmat, migrateout_cdmat, migrateback_cdmat, stray_cdmat, disperseLocal_cdmat, correlation_matrix, subpopmort_file) each accept, per batch, ONE of: a single raw R matrix; a file path; or sometimes "N". When providin a matrix as an R variable, the cdclimate and per-sex options are not available. To vary a matrix over time (the | cdclimate mechanism) or by sex (~), you must supply file *paths*, not matrices (e.g. pv$mate_cdmat <- "surface0.csv|surface5.csv").

When a matrix is supplied, it is written to a csv by launch_cdmetapop() (named after the R variable; see the package's file organization documentation). If you want control over a matrix file's location/format, supply a path and provide the file separately.

Value

An R6 PopVars object.

Examples

```
# 1 batch, defaults taken from PopVars.csv's first row:
mypopvars <- PopVars(n_batches = 1)

# Default 4-batch object, matching PopVars.csv:
mypopvars <- PopVars()


# Edit a column in place (one value per batch):
mypopvars$matemoveno <- c(6, 6, 4, 6)

# Add a batch (copies the last row; edit afterward):
mypopvars$add_row()
# or equivalently:
add_rows(mypopvars)

# Build from an existing PopVars csv, overriding one column:
## Not run:
mypopvars <- PopVars(path = "popvars/PopVars.csv", matemoveno = 6)

## End(Not run)
```

---

read_cdmetapop *Read an existing CDMetaPOP input file into an R6 object*

Description

Read an existing CDMetaPOP input file into an R6 object

Usage

```
read_cdmetapop(filepath, type = "RunVars")
```

Arguments

filepath	Path to an existing CDMetaPOP input csv.
type	Which input file type filepath is. One of "RunVars" (the default), "PopVars", "PatchVars", "ClassVars", or "DiseaseVars". Case-insensitive.

Value

An R6 object of the corresponding class (e.g. a RunVars() object).

Examples

```
## Not run:
myrunvars <- read_cdmetapop("RunVars.csv", type = "RunVars")
mypopvars <- read_cdmetapop("popvars/PopVars.csv", type = "PopVars")
mypatchvars <- read_cdmetapop("patchvars/PatchVarsS1.csv", type = "PatchVars")
myclassvars <- read_cdmetapop("classvars/ClassVars_AS1.csv", type = "ClassVars")
mydiseasevars <- read_cdmetapop("otherfiles/disease/DiseaseVars_SIR.csv", type = "DiseaseVars")

## End(Not run)
```

RunVars *Create a RunVars object*

Description

Constructs an R6 object representing a CDMetaPOP RunVars.csv input file – the top of CDMetaPOP's input-file hierarchy. Each row is a separate simulation "run", and its Popvars field references one or more PopVars() objects/files. CDMetaPOP executes every run x every PopVars batch x every Monte Carlo replicate (mcruns) as a separate simulation. Columns may be edited as a whole after construction, e.g. myrunvars$runtime <- c(5, 5, 10). **To build the PopVars object from an existing PopVars file, use the 'path' argument or the read_cdmetapop() function.

Usage

```
RunVars(
  n_runs = 4,
  Popvars = c("popvars/PopVars.csv", "popvars/PopVars.csv",
    "popvars/PopVars_IntroducePopulation.csv", "popvars/PopVars_Climate.csv"),
  sizecontrol = c("Y", "N", "N", "N"),
  constMortans = c("1", "1", "1", "1"),
  mcruns = c("1", "1", "1", "2"),
  runtime = c("5", "5", "5", "5"),
  output_years = c("1", "1", "1", "1"),
  gridformat = c("cdpop", "cdpop", "cdpop", "cdpop"),
  gridsampling = c("Sample", "Sample", "Sample", "Sample"),
  summaryOutput = c("N", "N", "N", "N"),
  cdclimgentime = c("0|1|2", "0|1|2", "0|5", "0|4|8"),
  startcomp = c("0", "0", "0", "0"),
  implementcomp = c("Back", "Back", "Back", "Back"),
  ncores = c("1", "1", "1", "2"),
  resolve_popvars = TRUE,
  path = NULL
)
```

Arguments

n_runs	Number of runs (rows). Defaults to 4, matching RunVars.csv. Use add_row() to add runs after construction.
Popvars	A PopVars() object, object-name/path string, or ;-joined/list of several (one per species), for each run.
sizecontrol	"Y" (body size drives size-linked processes) or "N" (age drives them).
constMortans	How constant mortality events compound: 1 (additive / mutually exclusive) or 2 (multiplicative / independent).
mcruns	Number of Monte Carlo replicates (positive integer).
runtime	Simulation run time in generations/years (positive integer).
output_years	Time steps to write output: a single integer (e.g. 2 -> years 0, 2, 4, ...), or exact years joined by 'l' (e.g. "0|3|4").
gridformat	Genotype output format: "cdpop", "general", "genalex", "structure", or "genepop".
gridsampling	"N" (output at natal grounds) or "Sample" (also output when away from natal grounds).
summaryOutput	"Y"/"N": produce per-patch summary metrics.
cdclimgentime	CDClimate module switch times: "0" (one surface throughout), or years, delimited with 'l' (e.g. "0|5|10").
startcomp	Year Lotka-Volterra competition begins (non-negative integer, multispecies runs only).
implementcomp	When to implement competition: "Back", "Out", or "N".
ncores	Number of cores for parallel Monte Carlo processing (positive integer).
resolve_popvars	If TRUE, strings entered for PopVars will be checked against the global environment for PopVars objects of the same name (primarily helpful for multispecies structure e.g., "popvars1;popvars2"). If FALSE, PopVars entries must be PopVars objects or they will be treated as filepaths.
path	Optional path to an existing RunVars csv file. If supplied, the object is built from that file, with the number of runs taken from its row count (so n_runs cannot also be given). Any column arguments supplied alongside path override the file's values, validated exactly as a later $ assignment would be. Values read from the file are always treated as literal file paths, so resolve_popvars applies only when path is NULL. With no overrides, this is equivalent to read_cdmetapop(path, type = "RunVars").

Details

Any column argument left unsupplied defaults to the corresponding values in example_files/RunVars.csv, matched to the last row in the example file.

CDMetaPOP reads RunVars.csv **headers as dictionary keys** (like PopVars.csv, unlike ClassVars.csv/PatchVars.csv) so every column name is used verbatim as both the data frame column name and this function's argument.

Delimiters: output_years and cdclimgentime support temporal variation over time with | (see user manual for more details).

Popvars **is an object-or-path field:** it accepts a PopVars() object directly, a file path to an existing PopVars.csv file, the name of a PopVars object in .GlobalEnv, or, for a multispecies run, several of any of these joined with ; (e.g. "popvars/PopVarsS1.csv;popvars/PopVarsS2.csv"), or a list of them.

Value

An R6 RunVars object.

Examples

```
# 1 run, defaults taken from RunVars.csv's first row:
myrunvars <- RunVars(n_runs = 1)

# Default 4-run object, matching RunVars.csv:
myrunvars <- RunVars()

# Edit a column in place (one value per run):
myrunvars$runtime <- c(5, 5, 10, 10)

# Add a run (copies the last row; edit afterward):
myrunvars$add_row()
# or equivalently:
add_rows(myrunvars)

# Build from an existing RunVars csv, overriding one column:
## Not run:
myrunvars <- RunVars(path = "RunVars.csv", runtime = 10)

## End(Not run)
```

---

summary_class *Plot CDMetaPOP Class Summary Dynamics*

Description

Summarizes and plots data from summary_classAllTime.csv files. Inputs can be a data frame, one file path, multiple file paths, or a CDMetaPOP output directory containing summary_classAllTime.csv files.

Usage

```
summary_class(
  data,
  type = "age_class",
  batch_labels = NULL,
  show_mc = TRUE,
```

```
  show_ci = TRUE,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  ...
)
```

Arguments

data	A dataframe, file path (.csv, .rds, .RData), vector of file paths, or a directory containing summary_classAllTime.csv files.
type	String specifying the plot type: "age_class" or "age_plus_one".
batch_labels	Optional named character vector used to relabel faceted source groups. Names should match folder names such as N.out1776184541 and values should be the labels displayed in the facet strips.
show_mc	Logical. If TRUE, plot individual Monte Carlo trajectories when multiple source files are supplied for line-based plots. Defaults to TRUE.
show_ci	Logical. If TRUE, plot the mean and a 95% confidence band across Monte Carlo replicates when multiple source files are supplied for line-based plots. Defaults to TRUE.
run	Integer run index used when data is a directory. Defaults to 0. Use "all" to include all runs.
batch	Integer batch index used when data is a directory. Defaults to 0. Use "all" to include all batches.
mc	Integer Monte Carlo index used when data is a directory. Defaults to 0. Use "all" to include all Monte Carlo replicates.
species	Integer species index used when data is a directory. Defaults to 0. Use "all" to include all species.
...	Additional arguments passed to specific plot types. Use n to choose the year interval for "age_class" plots.

Value

A ggplot object.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")

summary_class(ex_dir, type = "age_class", n = 10)
summary_class(ex_dir, type = "age_plus_one")
```

summary_dataframe *Return CDMetaPOP Output Data as Data Frames*

Description

Reads CDMetaPOP population, class, disease-state, or individual output files and returns data frames for custom plotting or downstream summaries.

Usage

```
summary_dataframe(
  data,
  type = c("pop", "class", "disease", "ind"),
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  years = NULL,
  file_type = "ind",
  patches = "all",
  state_names = NULL,
  cumulative_states = NULL,
  state_column = "States_SecondUpdate",
  disease_format = c("long", "wide"),
  summary_format = c("long", "wide")
)
```

Arguments

data	A data frame, file path, vector of file paths, run directory, or top-level CDMetaPOP output directory.
type	Character. Which file type to read: "pop" for summary_popAllTime.csv, "class" for summary_classAllTime.csv, "disease" for summary_popAllTime_DiseaseStates.csv, or "ind" for ind##.csv / ind##_Sample.csv files.
run	Integer run index used when data is a directory. Defaults to 0. Use "all" to include all runs.
batch	Integer batch index used when data is a directory. Defaults to 0. Use "all" to include all batches.
mc	Integer Monte Carlo index used when data is a directory. Defaults to 0. Use "all" to include all Monte Carlo replicates.
species	Integer species index used when data is a directory. Defaults to 0. Use "all" to include all species.
years	Optional integer vector of years/generations to include for type = "ind".
file_type	Character. Which individual file type to read for type = "ind". Use "ind" for ind##.csv files or "ind_Sample" for ind##_Sample.csv files. Defaults to "ind".

patches — Patch IDs to include for type = "ind". Use "all" to include all patches, a single patch ID, or a vector/range of patch IDs. Defaults to "all".

state_names — Character vector used to name disease states for type = "disease". If NULL, states are named numerically.

cumulative_states — Character vector of disease state names to calculate as running totals for type = "disease".

state_column — Character. Disease-state column to parse for type = "disease". Defaults to "States_SecondUpdate".

disease_format — Character. Use "long" to return one row per year/source/state, or "wide" to return one column per state. Defaults to "long".

summary_format — Character. Use "long" to split pipe-delimited summary_popAllTime.csv and summary_classAllTime.csv columns into one row per patch or class while keeping metrics in separate columns. Use "wide" to return the raw summary columns with metadata. Defaults to "long".

Value

A data frame with source metadata columns. For type = "pop" and type = "class", the default long format includes PatchID or ClassID indexing values split from pipe-delimited metric columns. For type = "disease", the default long format includes State and Count columns.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")

pop_df <- summary_dataframe(ex_dir, type = "pop")
class_df <- summary_dataframe(ex_dir, type = "class")
disease_df <- summary_dataframe(ex_dir, type = "disease")
ind_df <- summary_dataframe(ex_dir, type = "ind", years = 9)

all_mc_pop_df <- summary_dataframe(ex_dir, type = "pop", mc = "all")
```

---

summary_disease *Summarize CDMetaPOP Disease States*

Description

Scans CDMetaPOP output directories, pulls state totals from summary_popAllTime_DiseaseStates.csv files, and returns a faceted ggplot comparing disease states across batches.

Usage

```
summary_disease(
  base_path,
  state_names = NULL,
  scenario_names = NULL,
```

```
  cumulative_states = NULL,
  state_column = "States_SecondUpdate"
)
```

Arguments

base_path	Character. Path to a CDMetaPOP output directory containing run#batch#mc#species# folders, or a parent directory containing those folders.
state_names	Character vector. Optional labels for disease states. If NULL, states are named numerically ("1", "2", "3", ...).
scenario_names	Character vector. Optional labels for batches. The first value labels batch 0, the second labels batch 1, and so on.
cumulative_states	Character vector. Names of states to calculate as running totals within each Monte Carlo replicate.
state_column	Character. Disease-state column to summarize. Defaults to "States_SecondUpdate".

Value

A ggplot object.

Examples

```
ex_dir <- system.file(
  "extdata",
  "Example_dat",
  package = "cdmetapopR"
)

summary_disease(ex_dir)

summary_disease(
  ex_dir,
  state_names = c("Susceptible", "Infected", "Recovered"),
  scenario_names = c("Batch 0", "Batch 1"),
  cumulative_states = "Recovered"
)
```

---

summary_ind *Plot CDMetaPOP Individual File Summaries*

Description

Plots simple summaries from CDMetaPOP ind##.csv or ind##_Sample.csv files. Inputs can be a data frame, one individual file, multiple individual files, a single run folder, or a top-level CDMetaPOP output directory containing run folders.

Usage

```
summary_ind(
  path,
  type = "age",
  year = 0,
  years = NULL,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  file_type = "ind",
  patches = "all",
  bins = 30
)
```

Arguments

path	A dataframe, file path, vector of file paths, run directory, or top-level output directory containing individual files.
type	String specifying the plot type: "cdist", "hindex", "age", "size", "age_size", or "movement".
year	Integer. Year/generation to plot for one-year plots. Defaults to 0.
years	Integer vector. Years/generations to include for movement plots. If NULL, all discovered years are used for "movement".
run	Integer. Run index used when path is a directory. Defaults to 0.
batch	Integer. Batch index used when path is a directory. Defaults to 0.
mc	Integer. Monte Carlo index used when path is a directory. Defaults to 0.
species	Integer. Species index used when path is a directory. Defaults to 0.
file_type	Character. Which individual file type to read. Use "ind" for ind##.csv files or "ind_Sample" for ind##_Sample.csv files. Defaults to "ind".
patches	Patch IDs to include. Use "all" to include all patches, a single patch ID such as 5, or a vector/range such as c(1, 3, 8) or 1:20. Defaults to "all".
bins	Integer. Number of bins for continuous histograms. Defaults to 30.

Value

A ggplot object.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")

summary_ind(ex_dir, type = "age", year = 9)
summary_ind(ex_dir, type = "age_size", year = 9)
summary_ind(ex_dir, type = "movement", years = 0:9, batch = 1, mc = 1)
```

summary_patch_map *Map CDMetaPOP Patch Abundance*

Description

Creates a faceted patch map from CDMetaPOP `ind##.csv` or `ind##_Sample.csv` files. Points are drawn at patch coordinates and scaled by the number of individuals in each patch.

Usage

```
summary_patch_map(
  path,
  type = "abundance",
  years = 0,
  run = 0,
  batch = 0,
  mc = 0,
  species = 0,
  file_type = "ind",
  patches = "all",
  states = NULL,
  facet_by_state = FALSE,
  locus = NULL,
  allele = NULL,
  metric = "Ho",
  labels = FALSE,
  log_scale = FALSE,
  crs = NULL
)
```

Arguments

path	A dataframe, file path, vector of file paths, run directory, or top-level output directory containing individual files.
type	Character. What to map: `"abundance"`, `"allele_frequency"`, or `"heterozygosity"`. Defaults to `"abundance"`.
years	Integer vector. Years/generations to map. Defaults to `0`.
run	Integer. Run index used when `path` is a directory. Defaults to `0`.
batch	Integer. Batch index used when `path` is a directory. Defaults to `0`.
mc	Integer. Monte Carlo index used when `path` is a directory. Defaults to `0`.
species	Integer. Species index used when `path` is a directory. Defaults to `0`.
file_type	Character. Which individual file type to read. Use `"ind"` for `ind##.csv` files or `"ind_Sample"` for `ind##_Sample.csv` files. Defaults to `"ind"`.
patches	Patch IDs to include. Use `"all"` to include all patches, a single patch ID such as `5`, or a vector/range such as `c(1, 3, 8)` or `1:20`. Defaults to `"all"`.

states	Optional vector of disease states to include. If NULL, all individuals are used.
facet_by_state	Logical. If TRUE, facet by disease state as well as year. Defaults to FALSE.
locus	Locus to map for genetic summaries, such as "L0" or 0. Required when type is "allele_frequency" or "heterozygosity".
allele	Allele to map for type = "allele_frequency", such as "A1" or 1.
metric	Heterozygosity metric to map for type = "heterozygosity". Use "Ho" or "He". Defaults to "Ho".
labels	Logical. If TRUE, add patch ID labels to the map. Defaults to FALSE.
log_scale	Logical. If TRUE, use log1p-transformed abundance for point sizes when type = "abundance". Defaults to FALSE.
crs	Optional coordinate reference system label. For example, use 5070 or "EPSG:5070" for NAD83 / Conus Albers. Coordinates are not transformed; this is used for labeling and stored as plot data metadata.

Value

A ggplot object with patch-level summary data in plot$data.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")

summary_patch_map(ex_dir, years = c(0, 5, 9), crs = 5070)
summary_patch_map(ex_dir, years = c(0, 5, 9), states = 1, crs = 5070)
summary_patch_map(ex_dir, years = c(0, 9), states = c(0, 1), facet_by_state = TRUE, crs = 5070)
summary_patch_map(ex_dir, type = "allele_frequency", years = 9, locus = "L0", allele = "A1")
summary_patch_map(ex_dir, type = "heterozygosity", years = 9, locus = "L0", metric = "Ho")
```

---

summary_pop *Plot CDMetaPOP Population Summary Dynamics*

Description

Summarizes and plots data from summary_popAllTime.csv files. Inputs can be a data frame, one file path, multiple file paths, or a CDMetaPOP output directory containing summary_popAllTime.csv files.

Usage

```
summary_pop(
  data,
  type = "N_initial",
  batch_labels = NULL,
  show_mc = TRUE,
  show_ci = TRUE,
  run = 0,
```

```
  batch = 0,
  mc = 0,
  species = 0,
  ...
)
```

Arguments

data	A dataframe, file path (.csv, .rds, .RData), vector of file paths, or a directory containing summary_popAllTime.csv files.
type	String specifying the plot type: "N_initial", "sex", "mature", "births", "myy_ratio", "patch", "allelic_richness", or "het".
batch_labels	Optional named character vector used to relabel faceted source groups. Names should match folder names such as N.out1776184541 and values should be the labels displayed in the facet strips.
show_mc	Logical. If TRUE, plot individual Monte Carlo trajectories when multiple source files are supplied. Defaults to TRUE.
show_ci	Logical. If TRUE, plot the mean and a 95% confidence band across Monte Carlo replicates when multiple source files are supplied. Defaults to TRUE.
run	Integer run index used when data is a directory. Defaults to 0. Use "all" to include all runs.
batch	Integer batch index used when data is a directory. Defaults to 0. Use "all" to include all batches.
mc	Integer Monte Carlo index used when data is a directory. Defaults to 0. Use "all" to include all Monte Carlo replicates.
species	Integer species index used when data is a directory. Defaults to 0. Use "all" to include all species.
...	Additional arguments passed to specific plot types. Use include_yys = TRUE for "sex" or "mature" plots, or years for "patch" plots.

Value

A ggplot object.

Examples

```
ex_dir <- system.file("extdata", "Example_dat", package = "cdmetapopR")

summary_pop(ex_dir, type = "N_initial")
summary_pop(ex_dir, type = "sex")
summary_pop(ex_dir, type = "mature", include_yys = TRUE)
summary_pop(ex_dir, type = "allelic_richness")
summary_pop(ex_dir, type = "het")
```

`write_cdmetapop` *Write a cdmetapopR input-file object to a csv file*

Description

Writes a single `ClassVars()`, `PatchVars()`, `PopVars()`, `RunVars()`, or `DiseaseVars()` object to a csv file in CDMetaPOP's input format: the same headers, column order, and unquoted values that `launch_cdmetapop()` writes.

Usage

```
write_cdmetapop(x, file, overwrite = FALSE)
```

Arguments

`x`	A `ClassVars()`, `PatchVars()`, `PopVars()`, `RunVars()`, or `DiseaseVars()` object.
`file`	File name or path of the csv to write. A bare file name (e.g. `"myclassvars.csv"`) is written to the working directory. Include the `.csv` extension. The containing directory must already exist.
`overwrite`	If `FALSE` (the default), an existing file at `file` is not replaced and an error is raised instead. Set to `TRUE` to replace it.

Details

This is a convenience for writing one object at a time. It is not needed to run a simulation, since `launch_cdmetapop()` writes every input file automatically. Only `x` itself is written:

- References stored as file paths (e.g. a `class_vars` value of `"classvars/ClassVars_AS1.csv"`, or a cost-distance matrix path) are written exactly as entered. They are not checked, and the files they point to are not written or copied.
- A reference cell holding a live R object – a `PopVars()`, `PatchVars()`, `ClassVars()`, or `DiseaseVars()` object, or a raw `matrix` – has no file path to write, so that cell is written as `NA` and a warning lists the affected cells. To keep the reference, write that object or matrix to its own file first and assign its path instead.
- A `PatchVars()` object's eight disease columns are written only if at least one of them is set, matching the 50-or-58 column file CDMetaPOP expects (see `PatchVars()`).

Value

`file`, invisibly.

Examples

```
myclassvars <- ClassVars()
out <- file.path(tempdir(), "myclassvars.csv")
write_cdmetapop(myclassvars, file = out)

# The written file can be read back in:
read_cdmetapop(out, type = "ClassVars")
```

Index

add_rows, 2
add_rows(), *7*, *11*, *20*, *26*
age_structure_proportions, 3
allele_frequencies_ind, 4
as.data.frame.ClassVars, 5

base::as.data.frame(), *5*

cdmetapop_to_gene, 6
ClassVars, 6
ClassVars(), *2*, *5*, *20*, *23*, *41*
create_cdmat, 9

DiseaseVars, 10
DiseaseVars(), *3*, *5*, *21, 22*, *41*

heterozygosity_ind, 12

launch_cdmetapop, 13
launch_cdmetapop(), *29*, *41*

make_classvars, 15
make_diseasevars, 15
make_patchvars, 16
make_popvars, 16
make_runvars, 17

pairwise_fst_ind, 17
PatchVars, 18
PatchVars(), *2*, *5*, *10, 11*, *23*, *26*, *28*, *41*
PopVars, 23
PopVars(), *2*, *5*, *11*, *21, 22*, *30–32*, *41*

read_cdmetapop, 29
read_cdmetapop(), *10*
RunVars, 30
RunVars(), *2*, *5*, *13, 14*, *30*, *41*

summary_class, 32
summary_dataframe, 34
summary_disease, 35
summary_ind, 36
summary_patch_map, 38
summary_pop, 39
summary_pop(), *13, 14*

write_cdmetapop, 41