Package: leapfrog (via r-universe)

February 26, 2026

Title Multistate Population Projection Model for Demographic and HIV Estimation

Version 0.1.8

Description Leapfrog is a multistate population projection model for estimating population, demographic indicators, and HIV epidemic. The model combines a standard cohort component model of population projection (CCMPP) with a multistate model for HIV infection, disease progression, and treatment. Statistical tools are implemented for joint inference from multiple demographic and epidemiologic data sources.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/hivtools/leapfrog

BugReports https://github.com/hivtools/leapfrog/issues

Depends R (>= 4.3.0)

Config/testthat/edition 3

Additional_repositories https://mrc-ide.r-universe.dev

LinkingTo Rcpp

Imports beers, dplyr, hdf5r, magrittr, Rcpp

Suggests abind, docopt, devtools, knitr, plyr, RcppEigen, readxl, reshape2, rmarkdown, SpectrumUtils, testthat (>= 3.0.0), withr, xml2

Remotes rlglaubius/SpectrumUtils,

VignetteBuilder knitr

Config/pak/sysreqs libhdf5-dev

Repository https://mrc-ide.r-universe.dev

Date/Publication 2026-02-26 11:43:03 UTC

RemoteUrl https://github.com/hivtools/leapfrog

RemoteRef main

RemoteSha 36ab3d2337a8f1f8fd004bb523dbb1db16556d25

RemoteSubdir leapfrogr

Contents

`beers_open_ended` *Beers Ordinary Graduation with Open-Ended Age Group*

Description

Graduates vector from 5-year to single-year age, assuming the final value is an open-ended age group rather than an additional 5-year age group to be graduated.

Usage

```
beers_open_ended(x)
```

Arguments

x	vector of length at least 6

get_leapfrog_ss *Get the state space dimensions for a particular model configuration*

Description

Get the state space dimensions for a particular model configuration

Usage

```
get_leapfrog_ss(configuration)
```

Arguments

configuration	The configuration to get state space for, see list_model_configurations.

Value

State space as a named list

get_state_space *Run leapfrog model fit*

Description

Run leapfrog model fit

Usage

```
get_state_space(configuration = "HivFullAgeStratification")
```

Arguments

configuration	The model configuration to run, see `list_model_configurations()` for available configurations

Value

The state space for this model configuration.

Examples

```
s <- get_state_space("HivCoarseAgeStratification")
```

get_time_slice *Slice a single year from model state*

Description

Slice a single year from model state

Usage

```
get_time_slice(state, index)
```

Arguments

state	The model state with time dimension
index	The index of the time step you want to extract

Value

List of model outputs for the specified time step. Can be used as input state for `run_model_from_state()` and `run_model_single_year()`. All outputs will have 1 fewer dimension than input state.

list_model_configurations *List the avaialble model configurations*

Description

List the avaialble model configurations

Usage

```
list_model_configurations()
```

Value

List of available model configurations

process_parameters_to_cpp

Process parameters and convert from 1 based indexing in R to 0 based indexing in C++. Also add in any defaults/extra parameters, e.g. h_art_stage_dur.

Description

Process parameters and convert from 1 based indexing in R to 0 based indexing in C++. Also add in any defaults/extra parameters, e.g. h_art_stage_dur.

Usage

```
process_parameters_to_cpp(parameters)
```

Arguments

parameters	The list of parameters to feed into the model.

Value

List of parameters with 0 based indexing and defaults.

process_parameters_to_r

Process parameters and convert from 0 based indexing for C++ to 1 based indexing in R.

Description

Process parameters and convert from 0 based indexing for C++ to 1 based indexing in R.

Usage

```
process_parameters_to_r(parameters)
```

Arguments

parameters	List of parameters.

Value

List of parameters with 1 based indexing.

process_pjnz *Prepare inputs from Spectrum PJNZ*

Description

Prepare inputs from Spectrum PJNZ

Usage

```
process_pjnz(
  pjnz,
  use_coarse_age_groups = FALSE,
  extract_child_params = FALSE,
  bypass_adult = FALSE
)
```

Arguments

pjnz	path to PJNZ file
use_coarse_age_groups	use the coarse age stratification
extract_child_params	If TRUE, then child parameters required to run the child model will be extracted. If FALSE, only adult parameters will be extracted from the PJNZ.
bypass_adult	produce parameters that will bypass the adult model when running the child model variant

Value

list of input parameters

Examples

```
pjnz <- system.file(
  "pjnz/bwa_aim-adult-art-no-special-elig_v6.13_2022-04-18.PJNZ",
  package = "leapfrog")
parameters <- process_pjnz(pjnz)
```

`read_parameters` *Read parameters from HDF5 file format. This implicitly processes the parameters from C++ 0 based indexing to R 1 based indexing.*

Description

Read parameters from HDF5 file format. This implicitly processes the parameters from C++ 0 based indexing to R 1 based indexing.

Usage

```
read_parameters(file_path)
```

Arguments

`file_path`	HDF5 file to read

`run_model` *Run leapfrog model fit*

Description

Run leapfrog model fit

Usage

```
run_model(
  parameters,
  configuration = "HivFullAgeStratification",
  output_years = seq(1970, 2030)
)
```

Arguments

`parameters`	Projection parameters
`configuration`	The model configuration to run, see `list_model_configurations()` for available configurations
`output_years`	Which years of the model to return from the simulation, defaults to all years from 1970 to 2030. Also used to control what years the simulation is run for. If output only 2030, simulation will be run from `projection_start_year` passed in the `parameters` list.

Value

List of model outputs, where the last dimension of each element is time, e.g. `p_totpop` state variable has dimensions 81 x 2. If `output_years` specified has length 61 then the `p_totpop` output will have dimensions 81 x 2 x 61.

Examples

```
pjnz <- system.file(
  "pjnz/bwa_aim-adult-art-no-special-elig_v6.13_2022-04-18.PJNZ",
  package = "leapfrog", mustWork = TRUE)
parameters <- process_pjnz(pjnz, use_coarse_age_groups = TRUE)
out <- run_model(parameters, "HivCoarseAgeStratification", 1970:2030)
```

`run_model_from_state` *Run leapfrog model fit from initial state*

Description

Run leapfrog model fit from initial state

Usage

```
run_model_from_state(
  parameters,
  configuration,
  initial_state,
  simulation_start_year,
  output_years = seq(1970, 2030)
)
```

Arguments

`parameters`	Projection parameters
`configuration`	The model configuration to run, see `list_model_configurations()` for available configurations
`initial_state`	The model will run from this initial state
`simulation_start_year`	Start the model simulation from a particular year
`output_years`	Which years of the model to return from the simulation, defaults to all years from 1970 to 2030. Also used to control what years the simulation is run for. If output only 2030, simulation will be run from `projection_start_year` passed in the `parameters` list.

Value

List of model outputs, where the last dimension of each element is time, e.g. `p_totpop` state variable has dimensions 81 x 2. If `output_years` specified has length 61 then the `p_totpop` output will have dimensions 81 x 2 x 61.

Examples

```
pjnz <- system.file(
  "pjnz/bwa_aim-adult-art-no-special-elig_v6.13_2022-04-18.PJNZ",
  package = "leapfrog", mustWork = TRUE)
parameters <- process_pjnz(pjnz, use_coarse_age_groups = TRUE)
out_first_half_years <- run_model(parameters, "HivCoarseAgeStratification", 1970:2000)
out_second_half_years <- run_model_from_state(
  parameters,
  "HivCoarseAgeStratification",
  get_time_slice(out_first_half_years, 31),
  2000,
  2001:2030)
```

---

run_model_single_year *Run leapfrog model fit for a single year*

Description

Run leapfrog model fit for a single year

Usage

```
run_model_single_year(
  parameters,
  configuration,
  initial_state,
  simulation_start_year
)
```

Arguments

parameters	Projection parameters
configuration	The model configuration to run, see list_model_configurations() for available configurations
initial_state	The model will run from this initial state
simulation_start_year	Start the model simulation from this year

Value

List of model outputs without the last time dimension. This is different from run_model_from_state() and run_model() that do include the last time dimension. Since only the next time step is returned, dropping the time dimensions makes it easier to feed the returned list into the next single year model run. In contrast to run_model_from_state() the p_totpop output will have dimensions 81 x 2 not 81 x 2 x 61.

Examples

```
pjnz <- system.file(
  "pjnz/bwa_aim-adult-art-no-special-elig_v6.13_2022-04-18.PJNZ",
  package = "leapfrog", mustWork = TRUE)
parameters <- process_pjnz(pjnz, use_coarse_age_groups = TRUE)
out_first_half_years <- run_model(parameters, "HivCoarseAgeStratification", 1970:2000)
prev_state <- get_time_slice(out_first_half_years, 31)
for (i in 2001:2029) {
 new_state <- run_model_single_year(parameters, "HivCoarseAgeStratification", prev_state, i)
  # Do things with new state, other processes, saving output etc.
  prev_state <- new_state
}
```

`save_parameters` *Save parameters to HDF5 file format. This implicitly processes the parameters to C++ 0 based indexing.*

Description

Save parameters to HDF5 file format. This implicitly processes the parameters to C++ 0 based indexing.

Usage

```
save_parameters(df, file_path)
```

Arguments

`df`	list/dataframe to serialize
`file_path`	where to save the HDF5 file

Index