Package: hestia (via r-universe)

October 20, 2026

Title Bayesian Compartmental Infection Models from Individual Outcomes

Version 0.1.0

Description Fit Bayesian compartmental infection models (such as susceptible-infected-recovered and multi-compartment hidden Markov variants) from individual-level outcome data. Models are composed from infection-process and observation-process components and fit using Stan via 'rstan'.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Biarch true

Depends R (>= 4.1.0)

Imports checkmate, dplyr, flexstanr (>= 0.2.0), methods, posterior, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.5.0), stats, tidyr

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Suggests bayesplot, ggplot2, knitr, lintr, patchwork, pkgload, rmarkdown, roxygen2, roxyglobals, spelling, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

URL https://accidda.github.io/hestia/, https://github.com/ACCIDDA/hestia

BugReports https://github.com/ACCIDDA/hestia/issues

LazyData true

LazyDataCompression xz

Config/roxyglobals/filename globals.R

Config/roxyglobals/unique FALSE

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Claire Perrin Smith [aut, cre] (ORCID: <https://orcid.org/0000-0003-1069-9460>), Jack Goodall [aut]

Maintainer Claire Perrin Smith <clairesmith6995@gmail.com>

Config/pak/sysreqs make

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

Date/Publication 2026-10-20 15:00:02 UTC

RemoteUrl https://github.com/cran/hestia

RemoteRef HEAD

RemoteSha 36831c95f08e6e43009346c9cd8951de44520953

Contents

hestia-package 2
get_transmission_details 3
make_infection_model 4
make_observation_model 5
progress 6
run_model 7
siir 9
siir_res 10
sir 10
sir_cov 11
sir_cov_res 12
sir_res 12
transmit 13

Index **14**

`hestia-package` *The 'hestia' package.*

Description

A DESCRIPTION OF THE PACKAGE

Author(s)

Maintainer: Claire Perrin Smith <clairesmith6995@gmail.com> (ORCID)

Authors:

- Claire Perrin Smith <clairesmith6995@gmail.com> (ORCID)
- Jack Goodall <jackwgoodall@hotmail.com>

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.7. https://mc-stan.org

See Also

Useful links:

- https://accidda.github.io/hestia/
- https://github.com/ACCIDDA/hestia
- Report bugs at https://github.com/ACCIDDA/hestia/issues

---

get_transmission_details

Create Transmission Probability Matrix

Description

Create Transmission Probability Matrix

Usage

```
get_transmission_details(inf_model)
```

Arguments

inf_model	infection process model object yielded by make_infection_model()

Value

A list with the following entries:

- "states", a character vector of state names
- "trans_matrix", a matrix with transition rate values between origin (column) and destination (row) compartments where given. Where no values are provided for transition rates the entry is equal to 1e-10.
- "mult_matrix", a matrix with split values for the transition between origin (column) and destination (row) compartments where given. Where no split values are provided for transition rates the entry is equal to 1.
- "trans_to_fit", a data frame with information on all transitions within the model, with parameter indexing details if the transition rate is being fit
- "mult_to_fit", a data frame with information on all splits specified in the model, with parameter indexing details if the split is being fit
- "inf_states", numeric values corresponding to the indices in "states" which are the infectious states
- "mult_ih_inf_probs", indicator for whether infectious states have shared or separate intahousehold infection probabilities. If FALSE then all infection probabilities are shared across infectious compartments.

Examples

```
# Split the infectious compartment into symptomatic (I_s) and asymptomatic
# (I_a) infections, with separate intra-household infection probabilities.
inf_model <- make_infection_model(
  transmit(from = "S",
           to = c("I_s", "I_a"),
           source = c("I_s", "I_a"),
           split = "phi"),
  progress(from = "I_s", to = "R", gamma_s = NA),
  progress(from = "I_a", to = "R", gamma_a = NA),
  mult_ih_inf_probs = TRUE)

get_transmission_details(inf_model)
```

`make_infection_model` *Build Infection Process Model*

Description

Build Infection Process Model

Usage

```
make_infection_model(..., mult_ih_inf_probs = FALSE, mult_eh_inf_probs = FALSE)
```

Arguments

`...` a series of progress or transmit function calls

`mult_ih_inf_probs` If FALSE then all intra-household infection probabilities are shared across infectious compartments. If TRUE, each infection compartment has a distinct probability.

`mult_eh_inf_probs` If FALSE then all extra-household infection probabilities are shared across infectious compartments. If TRUE, each infection compartment has a distinct probability.

Value

A data frame with a row for each unique transition in the infection process model and the following columns:

- "from", the name of the origin compartment
- "to", the name of the destination compartment
- "source", NA (so output columns match with `transmit`)

- "rate_name", name of transition rate parameter if fitting
- "rate_value", numeric value of transition rate if not fitting
- "split_name", name of split parameter if fitting
- "split_value", numeric value of split if not fitting
- "source", names of compartment(s) that are the source of new infections
- "mult_ih_inf_probs", a logical indicator for whether infectious compartments have separate (TRUE) or shared (FALSE) intra-household infection probabilities

Examples

```
# Basic SIR model with recovery rate gamma to be fit
make_infection_model(transmit(from = "S", to = "I"),
                     progress(from = "I", to = "R", gamma = NA))

# Split the infectious compartment into symptomatic (I_s) and asymptomatic
# (I_a) infections, with separate intra-household infection probabilities.
make_infection_model(
  transmit(from = "S",
           to = c("I_s", "I_a"),
           source = c("I_s", "I_a"),
           split = "phi"),
  progress(from = "I_s", to = "R", gamma_s = NA),
  progress(from = "I_a", to = "R", gamma_a = NA),
  mult_ih_inf_probs = TRUE)
```

---

make_observation_model

Make Observation Process Model

Description

Make Observation Process Model

Usage

```
make_observation_model(...)
```

Arguments

...	A series of named vectors. Each vector corresponds to an observation type. Each entry in the vector is named for the corresponding compartment in the infection process model. The value is the probability of observing a positive observation given the individual is in the compartment.

Value

A list with one entry for each observation type.

Examples

```
# Observation process for an SIR model with two observations, PCR test
# results and presence of IgG antibodies.
make_observation_model(
  pcr = c("S" = 0.05, "I" = 0.95, "R" = 0.05),
  igg = c("S" = 0.01, "I" = 0.01, "R" = 0.8)
)
```

`progress` *Define a non-infection transition*

Description

Defines a state transition in the infection process model which does not represent an transmission (infection) event

Usage

```
progress(from, to, split = NA, ...)
```

Arguments

`from`	string giving name of origin compartment
`to`	string or vector of strings giving the name of the destination compartment
`split`	an optional character or numeric vector indicating what proportion of individuals transition into each of the `to` compartments
`...`	Argument specifying the name and optionally the value of the transition rate parameter. For example if the parameter name is "gamma" and it is being fit this argument should be `gamma = NA`. If we want to provide a numeric value, x, for "gamma" this argument should be `gamma = x`.

Value

Data frame with a row for each unique transition and the following columns:

- "from", the name of the origin compartment
- "to", the name of the destination compartment
- "source", NA (so output columns match with `transmit`)
- "rate_name", name of transition rate parameter if fitting
- "rate_value", numeric value of transition rate if not fitting
- "split_name", name of split parameter if fitting
- "split_value", numeric value of split if not fitting

Examples

```
# Transition from compartment A to B which occurs at rate gamma where
# gamma is a user-specified value of 0.2
progress(from = "A", to = "B", gamma = 0.2)

# Transition from compartment A to B which occurs at rate gamma where
# gamma is fit
progress(from = "A", to = "B", gamma = NA)

# Transition from compartment A split into two destination compartments, B1
# and B2, were individuals leave A at rate delta (fit) and phi (fit) is the
# proportion of those who leave A who go into B1 (i.e. 1-phi go into B2).
# gamma is fit
progress(from = "A", to = c("B1", "B2"), split = "phi", delta = NA)
```

`run_model` *Run Household Transmission Model*

Description

Run Household Transmission Model

Usage

```
run_model(
  inf_model,
  obs_model,
  data,
  init_probs,
  t_start = 1,
  epsilon = 1e-10,
  ih_cov = NULL,
  eh_cov = NULL,
  init = NULL,
  save_llik = FALSE,
  save_states = FALSE,
  stan_opts = stan_options()
)
```

Arguments

`inf_model`	Infection process model object generated by make_infection_model
`obs_model`	Observation process model object generated by make_observation_model
`data`	data frame with columns the following columns: • "t", time of observation

- "hh_id", numeric household ID which must be 1, 2, ..., N where N is the total number of household
- "part_id", participant ID within the household which must be 1, 2, ... n where n is the number of observed household members
- "hh_size", number of household members (whether observed or not)
- At least one column with binary outcome information where the column name matches the outcome name provided to make_observation_model

`init_probs` vector of initial probabilities for each infection process model state

`t_start` integer value corresponding to the start of model time

`epsilon` very small number to use for zero probability transitions

`ih_cov` NULL for run without covariates, otherwise data frame with intra-household covariates for each participant

`eh_cov` NULL for run without covariates, otherwise data frame with extra-household covariates for each participant

`init` initial conditions for MCMC chains

`save_llik` whether to write the per-household log-likelihood (`llik_final`) in the model's generated quantities, for use with `loo`/`waic` (default `FALSE`). When `FALSE`, `llik_final` is length 0 and the extra per-household forward pass is skipped.

`save_states` whether to write the per-participant log forward probabilities (`logalpha`) in the model's generated quantities, for reconstructing latent state probabilities (default `FALSE`). When `FALSE`, `logalpha` is a 0x0 matrix.

`stan_opts` a named list of Stan sampler options, as produced by `stan_options()` (for example `stan_options(iter = 1000)`).

Value

`draws_array` object with chains for each model parameter

Examples

```
# Subset sir package data to 10 households (for speedy example)
sir_sub <- sir[sir$hh_id <= 10, ]

# Make infetion process model for SIR model
inf_mod <- make_infection_model(
  transmit(from = "S", to = "I"),
  progress(from = "I", to = "R", gamma = NA))

# Make observation process model
obs_mod <- make_observation_model(
  pcr = c("S" = 0.05, "I" = 0.95, "R" = 0.05),
  igg = c("S" = 0.01, "I" = 0.01, "R" = 0.8))

# Fitting the Stan model is slow, so it is not run on CRAN.

run_model(inf_model = inf_mod,
          obs_model = obs_mod,
```

```
data = sir_sub,
init_probs = c(1 - 2 * 1e-10, 1e-10, 1e-10))
```

`siir` *Simulated SIIR household data*

Description

Individual-level outcomes simulated from an SIIR model for 500 households, in which infection splits into symptomatic (`Is`) and asymptomatic (`Ia`) compartments. The two infectious compartments have separate intra-household infection probabilities. Each individual has three binary observations: one informative about active infection (e.g. a PCR test), one informative about recovery (e.g. an IgG antibody test), and one indicating symptoms.

Usage

```
siir
```

Format

A data frame with one row per individual per day and the following columns:

t Day of observation.

part_id Participant identifier within the household.

enroll Enrollment indicator (1 = enrolled, 0 = not enrolled).

state True latent compartment (1 = S, 2 = Is, 3 = Ia, 4 = R).

hh_size Household size.

hh_id Household identifier.

pcr Binary outcome informative about active infection.

igg Binary outcome informative about recovery.

symp Binary outcome indicating symptoms.

Source

Generated by `data-raw/siir_data.R`.

siir_res — *Fitted draws for the SIIR model*

Description

Posterior draws from `run_model()` fit to the siir data using an infection process with separate symptomatic and asymptomatic compartments. Bundling the result lets the vignette display model output without re-running the Stan model. Parameters are on the natural (model) scale: probabilities, rates, and the symptomatic split are returned in their original units rather than on the logit fitting scale.

Usage

```
siir_res
```

Format

A `posterior` `draws_array` with parameters `eh_prob`, `ih_prob_Is`, `ih_prob_Ia`, `gamma_s`, `gamma_a`, and `phi`.

Source

Generated by `data-raw/siir_res.R`.

sir — *Simulated SIR household data*

Description

Individual-level outcomes simulated from a basic susceptible-infected-recovered (SIR) model for 500 households. Everyone starts susceptible, the daily extra-household infection probability is 1%, and the daily intra-household infection probability is 5%. Each individual has two binary observations: one informative about active infection (e.g. a PCR test) and one informative about recovery (e.g. an IgG antibody test).

Usage

```
sir
```

Format

A data frame with one row per individual per day and the following columns:

t Day of observation.

part_id Participant identifier within the household.

enroll Enrollment indicator (1 = enrolled, 0 = not enrolled).

state True latent compartment (1 = S, 2 = I, 3 = R).

hh_size Household size.

hh_id Household identifier.

pcr Binary outcome informative about active infection.

igg Binary outcome informative about recovery.

Source

Generated by `data-raw/sir_data.R`.

`sir_cov` *Simulated SIR household data with covariates*

Description

Individual-level outcomes simulated from a basic SIR model for 500 households, together with two binary individual-level covariates that affect the intra- and extra-household infection probabilities. Used to demonstrate fitting covariate effects.

Usage

```
sir_cov
```

Format

A list with two elements:

observations A data frame of outcomes with the same columns as sir.

covariates A data frame with one row per individual and two binary covariate columns, `x1` and `x2`.

Source

Generated by `data-raw/sir_cov_data.R`.

sir_cov_res — *Fitted draws for the SIR model with covariates*

Description

Posterior draws from `run_model()` fit to sir_cov with both intra- and extra-household covariates. Bundling the result lets the vignette display model output without re-running the Stan model. Parameters are on the natural (model) scale: infection probabilities and rates in their original units and covariate coefficients exponentiated, rather than on the logit and log fitting scales.

Usage

```
sir_cov_res
```

Format

A `posterior` `draws_array` with parameters `eh_prob`, `ih_prob`, `gamma`, `x1_eh`, `x2_eh`, `x1_ih`, and `x2_ih`.

Source

Generated by `data-raw/sir_cov_res.R`.

sir_res — *Fitted draws for the basic SIR model*

Description

Posterior draws from `run_model()` fit to the sir data using a basic SIR infection process (fitting the recovery rate) and a two-test observation process. Bundling the result lets the vignette display model output without re-running the Stan model. Parameters are on the natural (model) scale: probabilities and rates are returned in their original units rather than on the logit fitting scale.

Usage

```
sir_res
```

Format

A `posterior` `draws_array` with parameters `eh_prob`, `ih_prob`, and `gamma`.

Source

Generated by `data-raw/sir_res.R`.

`transmit` *Define a infection transition*

Description

Defines a state transition in the infection process model which is the result of a transmission (infection) event

Usage

```
transmit(from, to, source = NA, split = NA)
```

Arguments

`from`	string giving name of origin compartment
`to`	string giving the name of the destination compartment
`source`	string (or vector of strings) designating which compartments are infectious. If NA, the destination compartment is presumed to be the infectious compartment.
`split`	an optional character or numeric vector indicating what proportion of individuals transition into each of the `to` compartments

Value

Data frame with a row for each unique transition and the following columns:

- "from", the name of the origin compartment
- "to", the name of the destination compartment
- "split_name", name of split parameter if fitting
- "split_value", numeric value of split if not fitting
- "source", names of compartment(s) that are the source of new infections i.e. the infectious compartments

Examples

```
# Specify S > E transition in an SEIR model
transmit(from = "S", to = "E", source = "I")

# Upon infection, individuals either enter Is (symptomatic infections) or
# Ia (asymptomatic infection) where 30% of infections are symptomatic.
transmit(from = "S", to = c("Is", "Ia"), split = 0.3)
```

Index

∗ **datasets**
    siir, 9
    siir_res, 10
    sir, 10
    sir_cov, 11
    sir_cov_res, 12
    sir_res, 12

get_transmission_details, 3

hestia *(*hestia-package*)*, 2
hestia-package, 2

make_infection_model, 4, *7*
make_observation_model, 5, *7, 8*

progress, *4*, 6

run_model, 7
run_model(), *10*, *12*

siir, 9, *10*
siir_res, 10
sir, 10, *11, 12*
sir_cov, 11, *12*
sir_cov_res, 12
sir_res, 12
stan_options(), *8*

transmit, *4*, 13