Package: seairmobility (via r-universe)

November 7, 2026

Type Package

Title Mobility-Based SEAIR Epidemic Models

Version 0.1.0

Description Tools for simulating, analysing, and fitting mobility-based SEAIR (Susceptible-Exposed-Asymptomatic-Infectious-Recovered) compartmental epidemic models with heterogeneous individual mobility. Each individual carries a fixed mobility trait that scales susceptibility and infectiousness through a rank-one kernel, extending the mobility-based compartmental framework of Jiang et al. (2025) <doi:10.1137/24M1691557> by adding a latent stage and a behavioural split between asymptomatic and symptomatic infectiousness. Provides a numerical solver for the underlying partial differential equation system, closed-form computation of the basic reproduction number R0 and the final epidemic size, and a parametric least-squares routine for recovering the mobility distribution from an observed aggregate symptomatic time series.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports deSolve (>= 1.30), stats, graphics

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Weinan Wang [aut, cre]

Maintainer Weinan Wang <ww@ou.edu>

Repository https://wwang-math.r-universe.dev

Date/Publication 2026-04-28 20:46:35 UTC

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

RemoteRef HEAD

RemoteSha c1161aaf46b6ea68f49681a82627e30ed303917c

Contents

seairmobility-package . . . 2
beta_mixture_density . . . 3
c_effective . . . 4
final_size . . . 4
final_size_general . . . 5
fit_mobility . . . 6
plot_seair . . . 8
R0_seair . . . 8
seair_aggregate . . . 9
seair_init . . . 10
seair_params . . . 11
seair_solve . . . 11

Index **13**

`seairmobility-package` *seairmobility: Mobility-Based SEAIR Epidemic Models*

Description

The **seairmobility** package provides tools for simulating and analysing mobility-based SEAIR compartmental epidemic models. Each individual in the population carries a fixed mobility trait $m \in (0, 1)$ that scales both their susceptibility and their infectiousness through a rank-one transmission kernel. The infectious period is split into an asymptomatic stage with relative infectiousness α and a symptomatic stage with mobility-reduction factor δ.

Details

The main functions are

`seair_params` Build a parameter list.

`seair_init` Build initial conditions.

`seair_solve` Solve the SEAIR system.

`seair_aggregate` Aggregate solution over mobility.

`R0_seair` Basic reproduction number.

`final_size` Final epidemic size under vanishing seed.

`fit_mobility` Parametric least-squares fit of the mobility distribution from an observed symptomatic time series.

Author(s)

Maintainer: Weinan Wang <ww@ou.edu>

References

Jiang, N., Chu, W., and Li, Y. (2025). Modeling, inference, and prediction in mobility-based compartmental models for epidemiology. *SIAM Journal on Applied Mathematics*, 85(5), 2355–2375. doi:10.1137/24M1691557.

`beta_mixture_density` *Beta-mixture density on [0, 1]*

Description

Evaluates a weighted mixture of Beta densities

$$f(m) \;=\; \sum_{k=1}^{K} w_k \,\mathrm{Beta}(m; a_k, b_k),$$

renormalised on $[0, 1]$. Weights are coerced to be nonnegative and to sum to one.

Usage

```
beta_mixture_density(m, weights, shape1, shape2)
```

Arguments

`m`	Numeric vector of finite evaluation points in $[0, 1]$.
`weights`	Numeric vector of mixture weights (length K).
`shape1`	Numeric vector of Beta a_k parameters.
`shape2`	Numeric vector of Beta b_k parameters.

Value

Numeric vector of densities at `m`.

Examples

```
m <- seq(0, 1, length.out = 101)
beta_mixture_density(m,
                     weights = c(0.5, 0.3, 0.2),
                     shape1  = c(2, 5, 9),
                     shape2  = c(8, 3, 2))
```

c_effective *Effective infectiousness duration of the SEAIR chain*

Description

Computes the scalar

$$c \;=\; \frac{\alpha}{\kappa + \gamma_A} \;+\; \frac{\delta\,\kappa}{(\kappa + \gamma_A)\,\gamma_I},$$

the integrated contribution of a unit of newly infected mass to the force of infection, collapsing the $E \to A \to I$ stage chain into a single number. The latency rate σ does not appear in c.

Usage

```
c_effective(params)
```

Arguments

params	A "seair_params" object.

Value

A positive scalar.

Examples

```
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
c_effective(pars)
```

final_size *Final epidemic size under the vanishing-seed regime*

Description

Solves the scalar fixed-point equation

$$M_R^{\infty} \;=\; \int_0^1 m f(m)\,(1 - e^{-\beta c M_R^{\infty} m})\,dm$$

for the first mobility moment of the recovered compartment at the end of the epidemic (with $I_0 \equiv 0$, $R_0 \equiv 0$, $S_0 = f$), and returns the total final epidemic size

$$R_{\infty} \;=\; 1 - \int_0^1 f(m)\,e^{-\beta c M_R^{\infty} m}\,dm.$$

When the basic reproduction number is less than or equal to one, the function returns $R_{\infty} = 0$.

Usage

```
final_size(params, f, m_grid = NULL, tol = 1e-10)
```

Arguments

params	A "seair_params" object.
f	A mobility density, either a function on $[0, 1]$ or a numeric vector of values on m_grid.
m_grid	Optional mobility grid used when f is numeric, or to evaluate the integrals (default: uniform grid of 201 points).
tol	Tolerance for uniroot.

Value

A list with R_inf (total final size), MR_inf (first moment), R0 (basic reproduction number), and converged (logical).

Examples

```
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
f <- function(m) 6 * m * (1 - m)
final_size(pars, f)
```

---

final_size_general *Final epidemic size under general proportional initial conditions*

Description

Solves the scalar equation of Theorem 4.3 (final-size relation),

$$M_R^\infty + \int_0^1 m S_0(m)\, \exp\!\big(-\beta m[c\,(M_R^\infty - M_R^0) + (\delta/\gamma_I - c)\, M_I^0]\big) dm \;=\; \int_0^1 m f(m)\, dm,$$

allowing a nonzero initial symptomatic fraction $I_0 = I_{\text{seed}} f$, with $E_0 = A_0 = R_0 = 0$.

Usage

```
final_size_general(params, f, m_grid = NULL, I_seed = 1e-04, tol = 1e-10)
```

Arguments

params	A "seair_params" object.
f	A mobility density, either a function on $[0, 1]$ or a numeric vector of values on m_grid.
m_grid	Optional mobility grid used when f is numeric, or to evaluate the integrals (default: uniform grid of 201 points).
I_seed	Initial symptomatic fraction.
tol	Tolerance for uniroot.

Value

A list with R_inf, MR_inf, R0, and converged.

Examples

```
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
f <- function(m) 6 * m * (1 - m)
final_size_general(pars, f, I_seed = 0.001)
```

---

fit_mobility *Fit a mobility distribution to an observed symptomatic time series*

Description

Fits a Beta-mixture mobility distribution f by minimising the sum of squared errors between an observed aggregate symptomatic time series $\langle I \rangle(t)$ and the simulated output of the mobility-based SEAIR model at the same times. All transmission and stage parameters are held fixed.

Usage

```
fit_mobility(
  times,
  I_obs,
  params,
  K = 3,
  m_grid = seq(0, 1, length.out = 51),
  I_seed = 1e-04,
  start = NULL,
  n_restarts = 5,
  control = list(maxit = 500),
  verbose = FALSE
)
```

Arguments

times	Numeric vector of observation times (starting at 0).
I_obs	Observed aggregate symptomatic fraction at each time.
params	A "seair_params" object holding fixed transmission and stage parameters.
K	Number of Beta-mixture components (default 3).
m_grid	Mobility grid for the solver.
I_seed	Initial symptomatic fraction in $[0, 1)$.
start	Optional finite user-supplied starting parameter vector of length $3K - 1$.
n_restarts	Number of random restarts (default 5).
control	List of optim control options.
verbose	Logical; if TRUE, print progress.

Details

The unconstrained optimisation parameter is, for each component $k = 1, \ldots, K$, a triple $(\log a_k, \log b_k, \log \tilde{w}_k)$ with mixture weights recovered as $w_k = \tilde{w}_k / \sum_j \tilde{w}_j$. The first weight's log is fixed to 0 to remove the scale ambiguity, so the parameter vector has length $3K - 1$.

Value

A list with weights, shape1, shape2 (the fitted Beta-mixture parameters), f_hat (density on m_grid), m_grid, loss (final SSE), I_fit (simulated aggregate symptomatic time series at times), and optim_result.

Examples

```
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
m <- seq(0, 1, length.out = 21)
f_true <- dbeta(m, 3, 3)
init <- seair_init(m, f_true, I_seed = 1e-4)
times <- seq(0, 20, by = 5)
sol <- seair_solve(init, pars, times)
I_obs <- seair_aggregate(sol)$I

fit <- fit_mobility(times, I_obs, pars,
                    K = 1, m_grid = m,
                    start = log(c(3, 3)),
                    n_restarts = 0,
                    control = list(maxit = 20))
fit$loss
```

plot_seair *Plot aggregate compartment trajectories*

Description

Convenience plot of the five aggregate compartment trajectories produced by a SEAIR solution.

Usage

```
plot_seair(sol, which = c("S", "E", "A", "I", "R"), ...)
```

Arguments

sol	A "seair_solution" object.
which	Character vector of compartments to plot; any subset of c("S","E","A","I","R").
...	Additional graphical parameters passed to matplot.

Value

Invisibly returns the aggregate data frame.

Examples

```
m <- seq(0, 1, length.out = 31)
f <- dbeta(m, 2, 2)
init <- seair_init(m, f, I_seed = 1e-4)
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
sol <- seair_solve(init, pars, times = seq(0, 30, by = 2))
plot_seair(sol, which = c("S","I","R"))
```

R0_seair *Basic reproduction number of the mobility-based SEAIR model*

Description

For a mobility distribution f on $(0, 1)$, the basic reproduction number at the disease-free equilibrium is

$$\mathcal{R}_0 \;=\; \beta\, c\, \langle m^2 f \rangle,$$

where c is as in c_effective and $\langle m^2 f \rangle = \int_0^1 m^2 f(m)\, dm$.

Usage

```
R0_seair(params, f, m_grid = NULL)
```

Arguments

`params`	A `"seair_params"` object.
`f`	A function returning the density $f(m)$ on $[0, 1]$, or a numeric vector of f values evaluated on `m_grid`.
`m_grid`	Mobility grid (used only if `f` is numeric).

Value

The basic reproduction number $\mathcal{R}_0$.

Examples

```
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)

## f given as a function (normalised automatically by the caller).
f <- function(m) 6 * m * (1 - m)    # Beta(2, 2) density on [0, 1]
R0_seair(pars, f)

## f given as values on a grid.
m <- seq(0, 1, length.out = 101)
R0_seair(pars, f(m), m_grid = m)
```

`seair_aggregate` *Aggregate a SEAIR solution over mobility*

Description

Computes the aggregate time series $\int_0^1 S(m,t)\,dm$, $\int_0^1 E(m,t)\,dm$, etc., using the trapezoidal rule on the solver's mobility grid.

Usage

```
seair_aggregate(sol)
```

Arguments

`sol`	A `"seair_solution"` object returned by `seair_solve`.

Value

A data frame with columns `time`, `S`, `E`, `A`, `I`, `R` (aggregate ratios).

Examples

```
m <- seq(0, 1, length.out = 31)
f <- dbeta(m, 2, 2)
init <- seair_init(m, f, I_seed = 1e-4)
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
sol <- seair_solve(init, pars, times = seq(0, 30, by = 2))
agg <- seair_aggregate(sol)
head(agg)
```

`seair_init` *Build initial conditions for the SEAIR system*

Description

Constructs initial profiles $(S_0, E_0, A_0, I_0, R_0)$ on the mobility grid from a user-supplied mobility density $f(m)$ and a seed fraction I_{seed} placed in the symptomatic compartment. The initial condition is "proportional": $S_0 = (1 - I_{\text{seed}})f$, $I_0 = I_{\text{seed}} f$, $E_0 = A_0 = R_0 = 0$.

Usage

```
seair_init(m_grid, f_vals, I_seed = 1e-04)
```

Arguments

`m_grid`	Numeric vector of mobility grid points in $[0, 1]$.
`f_vals`	Numeric vector of densities $f(m)$ evaluated at `m_grid`. It is renormalised internally so that the trapezoidal integral over `m_grid` equals 1.
`I_seed`	Non-negative fraction of the population seeded in the symptomatic compartment (default `1e-4`).

Value

A list of class `"seair_init"` with entries `m_grid`, `f`, `S`, `E`, `A`, `I`, `R`, and `I_seed`.

Examples

```
m <- seq(0, 1, length.out = 101)
f <- dbeta(m, 2, 2)
seair_init(m, f, I_seed = 1e-4)
```

seair_params *Build a parameter list for the mobility-based SEAIR model*

Description

Constructs and validates the list of transmission and stage-progression parameters used by the mobility-based SEAIR model $\partial_t S = -\beta m S \mathcal{K}$, $\partial_t E = \beta m S \mathcal{K} - \sigma E$, $\partial_t A = \sigma E - (\kappa + \gamma_A) A$, $\partial_t I = \kappa A - \gamma_I I$, $\partial_t R = \gamma_A A + \gamma_I I$, where $\mathcal{K}(A, I) = \int_0^1 \bar{m}\,(\alpha A(\bar{m}) + \delta I(\bar{m}))\, d\bar{m}$.

Usage

```
seair_params(beta, sigma, kappa, gamma_A, gamma_I, alpha = 1, delta = 1)
```

Arguments

`beta`	Transmission rate $\beta > 0$.
`sigma`	Rate $\sigma > 0$ of progression from latent (E) to asymptomatic (A).
`kappa`	Symptom-onset rate $\kappa > 0$ (A to I).
`gamma_A`	Recovery rate $\gamma_A > 0$ from the asymptomatic compartment.
`gamma_I`	Recovery rate $\gamma_I > 0$ from the symptomatic compartment.
`alpha`	Relative infectiousness $\alpha \in [0, 1]$ of the asymptomatic stage.
`delta`	Mobility-reduction factor $\delta \in (0, 1]$ for the symptomatic stage.

Value

A list of class `"seair_params"` containing the checked parameters.

Examples

```
seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
             gamma_A = 0.1, gamma_I = 0.13,
             alpha = 0.5, delta = 0.3)
```

seair_solve *Solve the mobility-based SEAIR system*

Description

Discretises the mobility-based SEAIR system on the user-supplied mobility grid (using the trapezoidal rule for the non-local integral $\mathcal{K}$) and integrates the resulting coupled ODE system over `times` with **deSolve**.

Usage

```
seair_solve(init, params, times, method = "lsoda", ...)
```

Arguments

init	A "seair_init" object from seair_init.
params	A "seair_params" object from seair_params.
times	Numeric vector of times at which the solution is reported (must be increasing and include 0).
method	Integration method passed to ode (default "lsoda").
...	Additional arguments passed to ode.

Value

A list of class "seair_solution" with entries times, m_grid, S, E, A, I, R (each a matrix of dimension length(times) by length(m_grid)), together with the input params and init.

Examples

```
m <- seq(0, 1, length.out = 31)
f <- dbeta(m, 2, 2)
init <- seair_init(m, f, I_seed = 1e-4)
pars <- seair_params(beta = 1.5, sigma = 0.3, kappa = 0.2,
                     gamma_A = 0.1, gamma_I = 0.13,
                     alpha = 0.5, delta = 0.3)
sol <- seair_solve(init, pars, times = seq(0, 30, by = 2))
```

Index

beta_mixture_density, 3

c_effective, 4, *8*

final_size, *2*, 4
final_size_general, 5
fit_mobility, *2*, 6

matplot, *8*

ode, *12*
optim, *7*

plot_seair, 8

R0_seair, *2*, 8

seair_aggregate, *2*, 9
seair_init, *2*, 10, *12*
seair_params, *2*, 11, *12*
seair_solve, *2*, *9*, 11
seairmobility *(*seairmobility-package*)*, 2
seairmobility-package, 2

uniroot, *5*, *6*