Package ‘seirMFG’

April 9, 2026

Type Package

Title Mean-Field Game Equilibrium for SEIR Epidemics on Networks

Version 0.1.0

Date 2026-04-02

Description Implements the forward-backward sweep algorithm for computing Nash equilibrium contact policies in SEIR epidemic mean-field games on heterogeneous contact networks, as described in Wang (2026) <doi:10.5281/zenodo.19381052>. Supports both heterogeneous networks with arbitrary degree distributions (e.g., truncated Poisson) and homogeneous networks. Computes equilibrium susceptible contact effort, value functions, epidemic trajectories, and the effective reproduction number Rt.

License MIT + file LICENSE

URL `https://doi.org/10.5281/zenodo.19381052`

Encoding UTF-8

Depends R (>= 4.0.0)

Imports stats, graphics

Suggests ggplot2, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation no

Author Weinan Wang [aut, cre]

Maintainer Weinan Wang <`ww@ou.edu`>

Repository CRAN

Date/Publication 2026-04-09 09:00:09 UTC

Contents

`compute_Rt` *Compute the time-varying effective reproduction number Rt*

Description

Computes R_t for the heterogeneous-network SEIR-MFG equilibrium using the next-generation spectral radius formula (Theorem 3.6 of Wang, 2026):

$$R_t = \frac{\beta}{\gamma\langle k\rangle}\sum_k k^2 P(k) n_k^{S*}(t) S_k(t).$$

This scalar formula holds for uncorrelated networks.

Usage

```
compute_Rt(result, network, gamma = 1)
```

Arguments

`result`	A list returned by `seir_mfg_het` or `sir_mfg_het`, containing matrices `S` and `nS`.
`network`	A list returned by `make_poisson_network`.
`gamma`	Numeric. Recovery rate. Default 1.

Value

A numeric vector of length $M + 1$ giving R_t at each time step.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
# Simple network setup (fast)
net <- make_poisson_network()


# Full solve then compute Rt (slow)
res <- seir_mfg_het(network = net, sigma = 2, verbose = FALSE)
Rt  <- compute_Rt(res, net)
plot(res$t_grid, Rt, type = "l", xlab = "Time", ylab = "Rt")
```

`make_poisson_network` *Build a truncated Poisson degree distribution*

Description

Constructs the degree distribution $P(k)$ for an uncorrelated Markovian network with a truncated Poisson degree distribution, and computes the transmission rate β required to achieve a target basic reproduction number $R_0 = \beta\langle k^2\rangle/(\gamma\langle k\rangle)$.

Usage

```
make_poisson_network(mu = 8, k_max = 25, R0 = 4, gamma = 1)
```

Arguments

`mu`	Numeric. Mean degree of the Poisson distribution. Default 8.
`k_max`	Integer. Maximum degree (truncation point). Default 25.
`R0`	Numeric. Target basic reproduction number. Default 4.
`gamma`	Numeric. Recovery rate. Default 1.

Value

A list with components:

k_vals Integer vector of degree values $1, \ldots, k\_max$.

PK Numeric vector of probabilities $P(k)$, summing to 1.

K_bar Mean degree $\langle k\rangle$.

K_bar2 Second moment $\langle k^2\rangle$.

beta Transmission rate achieving the target R_0.

R0 The target R_0.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
net <- make_poisson_network(mu = 8, k_max = 25, R0 = 4, gamma = 1)
net$beta   # approximately 0.4444
net$K_bar  # approximately 8
```

`seir_mfg_het` *Solve the heterogeneous-network SEIR mean-field game*

Description

Computes the Nash equilibrium contact policy and epidemic trajectory for the SEIR mean-field game on an uncorrelated heterogeneous network using the iterative forward-backward sweep (FBS) algorithm (Section 6.1 of Wang, 2026).

Usage

```
seir_mfg_het(network, sigma = 2, gamma = 1, r_I = 50, n_min = 0.1,
             eps = 1, C_E = 0, C_I = 0, S0 = 0.98, E0 = 0.01,
             I0 = 0.01, T_end = 30, dt = 0.05, omega = 0.3,
             tol = 1e-7, max_iter = 2000, verbose = TRUE)
```

Arguments

`network`	A list returned by `make_poisson_network`.
`sigma`	Numeric. Incubation rate. Default 2.
`gamma`	Numeric. Recovery rate. Default 1.
`r_I`	Numeric. One-time infection cost. Default 50.
`n_min`	Numeric. Minimum contact effort in (0,1). Default 0.1.
`eps`	Integer. Isolation cost exponent; $f_k(n) = k^\epsilon(1/n - 1)$. Default 1.
`C_E`	Numeric. Health cost in state E. Default 0.
`C_I`	Numeric. Health cost in state I. Default 0.
`S0`	Numeric. Initial susceptible fraction. Default 0.98.
`E0`	Numeric. Initial exposed fraction. Default 0.01.
`I0`	Numeric. Initial infectious fraction. Default 0.01.
`T_end`	Numeric. Time horizon. Default 30.
`dt`	Numeric. Time step. Default 0.05.
`omega`	Numeric. FBS relaxation parameter in (0,1]. Default 0.3.
`tol`	Numeric. Convergence tolerance. Default 1e-7.
`max_iter`	Integer. Maximum FBS iterations. Default 2000.
`verbose`	Logical. Print convergence info. Default TRUE.

Value

A list with components:

S, E, I, R Matrices of shape $K \times (M+1)$: compartment fractions.

nS Matrix of shape $K \times (M+1)$: equilibrium effort $n_k^{S*}(t)$.

I_agg Numeric vector: aggregate infectious fraction $\sum_k P(k) I_k(t)$.

t_grid Numeric vector of time points.

converged Logical. Whether FBS converged.

n_iter Integer. FBS iterations used.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
# Simple network setup (fast)
net <- make_poisson_network(mu = 8, k_max = 25, R0 = 4)


# Full FBS solve (slow: ~40 iterations x 25 degree classes)
res <- seir_mfg_het(network = net, sigma = 2, verbose = FALSE)
plot(res$t_grid, res$I_agg, type = "l", col = "red",
     xlab = "Time", ylab = "Aggregate I(t)")
```

`seir_mfg_hom` *Solve the homogeneous-network SEIR mean-field game*

Description

Computes the Nash equilibrium contact policy and epidemic trajectory for the SEIR mean-field game on a homogeneous network where all nodes have degree k and the effective per-contact transmission rate `beta_k` is fixed.

Usage

```
seir_mfg_hom(k = 8, sigma = 2, beta_k = 4, gamma = 1, r_I = 50,
             n_min = 0.1, eps = 1, C_E = 0, C_I = 0, S0 = 0.98,
             E0 = 0.01, I0 = 0.01, T_end = 30, dt = 0.05,
             omega = 0.3, tol = 1e-7, max_iter = 2000, verbose = TRUE)
```

Arguments

`k`	Integer. Node degree. Default 8.
`sigma`	Numeric. Incubation rate. Default 2.
`beta_k`	Numeric. Effective transmission rate βk. Default 4.
`gamma`	Numeric. Recovery rate. Default 1.
`r_I`	Numeric. One-time infection cost. Default 50.
`n_min`	Numeric. Minimum contact effort. Default 0.1.
`eps`	Integer. Isolation cost exponent. Default 1.
`C_E`	Numeric. Health cost in state E. Default 0.
`C_I`	Numeric. Health cost in state I. Default 0.
`S0`	Numeric. Initial susceptible fraction. Default 0.98.
`E0`	Numeric. Initial exposed fraction. Default 0.01.
`I0`	Numeric. Initial infectious fraction. Default 0.01.
`T_end`	Numeric. Time horizon. Default 30.
`dt`	Numeric. Time step. Default 0.05.
`omega`	Numeric. FBS relaxation parameter. Default 0.3.
`tol`	Numeric. Convergence tolerance. Default 1e-7.
`max_iter`	Integer. Maximum FBS iterations. Default 2000.
`verbose`	Logical. Print convergence info. Default TRUE.

Value

A list with components:

S, E, I, R Numeric vectors of length $M + 1$: compartment fractions.

nS Numeric vector: equilibrium effort $n^{S*}(t)$.

t_grid Numeric vector of time points.

converged Logical.

n_iter Integer. Iterations used.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
# Full FBS solve (slow: ~40 iterations)
res <- seir_mfg_hom(k = 8, sigma = 2, beta_k = 4, verbose = FALSE)
plot(res$t_grid, res$I, type = "l", xlab = "Time", ylab = "I(t)")
lines(res$t_grid, res$nS, col = "red", lty = 2)
legend("topright", c("I(t)", "nS*(t)"),
       col = c("black", "red"), lty = 1:2)
```

`sir_mfg_het` *Solve the heterogeneous-network SIR mean-field game*

Description

Computes the SIR-MFG Nash equilibrium on an uncorrelated heterogeneous network. This corresponds to the limit $\sigma \to \infty$ of `seir_mfg_het` (Lemma 5.7 of Wang, 2026). The initial infectious seed absorbs the exposed compartment: $I_0^{\mathrm{SIR}} = I_0 + E_0$.

Usage

```
sir_mfg_het(network, gamma = 1, r_I = 50, n_min = 0.1, eps = 1,
            C_I = 0, S0 = 0.98, I0_sir = 0.02, T_end = 30,
            dt = 0.05, omega = 0.3, tol = 1e-7,
            max_iter = 2000, verbose = TRUE)
```

Arguments

`network`	A list returned by `make_poisson_network`.
`gamma`	Numeric. Recovery rate. Default 1.
`r_I`	Numeric. One-time infection cost. Default 50.
`n_min`	Numeric. Minimum contact effort. Default 0.1.
`eps`	Integer. Isolation cost exponent. Default 1.
`C_I`	Numeric. Health cost in state I. Default 0.
`S0`	Numeric. Initial susceptible fraction. Default 0.98.
`I0_sir`	Numeric. Initial infectious fraction (absorbs exposed seed). Default 0.02.
`T_end`	Numeric. Time horizon. Default 30.
`dt`	Numeric. Time step. Default 0.05.
`omega`	Numeric. FBS relaxation parameter. Default 0.3.
`tol`	Numeric. Convergence tolerance. Default 1e-7.
`max_iter`	Integer. Maximum FBS iterations. Default 2000.
`verbose`	Logical. Print convergence info. Default TRUE.

Value

A list with components `S`, `I`, `R`, `nS`, `I_agg`, `t_grid`, `converged`, `n_iter`.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
# Simple network setup (fast)
net <- make_poisson_network()


# Full FBS solve (slow)
res <- sir_mfg_het(network = net, verbose = FALSE)
plot(res$t_grid, res$I_agg, type = "l", col = "blue",
     xlab = "Time", ylab = "Aggregate I(t)")
```

`sir_mfg_hom` *Solve the homogeneous-network SIR mean-field game*

Description

Computes the SIR-MFG Nash equilibrium on a homogeneous network. The initial infectious seed absorbs the exposed compartment: $I_0^{\mathrm{SIR}} = I_0 + E_0$.

Usage

```
sir_mfg_hom(k = 8, beta_k = 4, gamma = 1, r_I = 50, n_min = 0.1,
            eps = 1, C_I = 0, S0 = 0.98, I0_sir = 0.02, T_end = 30,
            dt = 0.05, omega = 0.3, tol = 1e-7,
            max_iter = 2000, verbose = TRUE)
```

Arguments

`k`	Integer. Node degree. Default 8.
`beta_k`	Numeric. Effective transmission rate βk. Default 4.
`gamma`	Numeric. Recovery rate. Default 1.
`r_I`	Numeric. One-time infection cost. Default 50.
`n_min`	Numeric. Minimum contact effort. Default 0.1.
`eps`	Integer. Isolation cost exponent. Default 1.
`C_I`	Numeric. Health cost in state I. Default 0.
`S0`	Numeric. Initial susceptible fraction. Default 0.98.
`I0_sir`	Numeric. Initial infectious fraction. Default 0.02.
`T_end`	Numeric. Time horizon. Default 30.
`dt`	Numeric. Time step. Default 0.05.
`omega`	Numeric. FBS relaxation parameter. Default 0.3.
`tol`	Numeric. Convergence tolerance. Default 1e-7.
`max_iter`	Integer. Maximum FBS iterations. Default 2000.
`verbose`	Logical. Print convergence info. Default TRUE.

Value

A list with components `S`, `I`, `R`, `nS`, `t_grid`, `converged`, `n_iter`.

References

Wang, W. (2026). Learning contact policies for SEIR epidemics on networks: A mean-field game approach. doi:10.5281/zenodo.19381052.

Examples

```
# Full FBS solve (slow)
res <- sir_mfg_hom(k = 8, beta_k = 4, verbose = FALSE)
plot(res$t_grid, res$I, type = "l", col = "blue",
     xlab = "Time", ylab = "I(t)")
```

Index