Package ‘socialSim’

October 30, 2025

Type Package

Title Simulate and Analyse Social Interaction Data

Version 0.1.6

Description Provides tools to simulate and analyse datasets of social interactions between individuals using hierarchical Bayesian models implemented in Stan. The package interacts with Stan via 'cmdstanr' (available from <https://mc-stan.org/r-packages/>) or 'rstan', depending on user setup. Users can generate realistic interaction data where individual phenotypes influence and respond to those of their partners, with control over sampling design parameters such as the number of individuals, partners, and repeated dyads. The simulation framework allows flexible control over variation and correlation in mean trait values, social responsiveness, and social impact, making it suitable for research on interacting phenotypes and on direct and indirect genetic effects ('DGEs' and 'IGEs'). The package also includes functions to fit and compare alternative models of social effects, including impact–responsiveness, variance–partitioning, and trait-based models, and to summarise model performance in terms of bias and dispersion. For more details on the study of social interactions and impact-responsiveness, see Moore et al. (1997) <doi:10.1111/j.1558-5646.1997.tb01458.x> and de Groot et al. (2022) <doi:10.1016/j.neubiorev.2022.104996>.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Depends R (>= 4.2)

Imports MASS, stats, future, future.apply,

Suggests rstan, cmdstanr, devtools, testthat, rmarkdown, knitr

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/RoriWijnhorst/socialSim

BugReports https://github.com/RoriWijnhorst/socialSim/issues

Additional_repositories https://stan-dev.r-universe.dev

NeedsCompilation no

Author Rori Efrain Wijnhorst [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8902-592X>)

Maintainer Rori Efrain Wijnhorst <roriwijnhorst@gmail.com>

Repository CRAN

Date/Publication 2025-10-30 14:20:02 UTC

Contents

run_model *Fit one of the available Stan models to simulated datasets*

Description

Fit one of the available Stan models to simulated datasets

Usage

```
run_model(sim, model = NULL, iter = 2000, seed = 1234, cores = 1)
```

Arguments

`sim`	Output from `simulate_data()`.
`model`	Name of the Stan model to use (choose from available options).
`iter`	Number of iterations per chain (default = 1000).
`seed`	Random seed for reproducibility.
`cores`	Number of CPU cores (used if cmdstanr or rstan is available).

Value

A list of fitted model summaries, one per dataset.

Examples

```
if (requireNamespace("cmdstanr", quietly = TRUE) ||
    requireNamespace("rstan", quietly = TRUE)) {
  sim <- simulate_data(ind = 100, Valpha = 0.2, Vepsilon = 0.1, iterations = 2)
  res <- run_model(sim, model = "Trait.stan", iter = 500, cores = 2)
  summary(res)
} else {
  message("CmdStanR or rstan not available; example skipped.")
}
```

simulate_data *Simulate social interaction datasets*

Description

This function generates datasets where individual phenotypes are influenced by both direct and indirect (social) effects, under a specified sampling design.

Usage

```
simulate_data(
  ind = 200,
  partners = 4,
  repeats = 1,
  iterations = 100,
  B_0 = 0,
  psi = NULL,
  Valpha,
  Vepsilon = NULL,
  Vpsi = 0,
  Vx = 1,
  Ve = 0.6,
  Vxe = 0,
  r_alpha_epsilon = 0,
  r_alpha_psi = 0,
  r_epsilon_psi = 0,
  r_alpha_x = 0,
  r_psi_x = 0,
  r_epsilon_x = 0,
  fix_total_var = TRUE
)
```

Arguments

`ind`	Number of individuals.
`partners`	Partners per individual.

`repeats` Repeats per unique dyad.

`iterations` Number of datasets to simulate.

`B_0` Population intercept.

`psi` Population-level responsiveness (social slope).

`Valpha` Direct effect (focal variance).

`Vepsilon` Indirect effect (partner variance).

`Vpsi` Social responsiveness (among individual variance in slopes).

`Vx` Partner trait variance.

`Ve` Residual variance.

`Vxe` Measurement error/within-individual variation in partner trait.

`r_alpha_epsilon`
Corr(alpha, epsilon).

`r_alpha_psi` Corr(alpha, psi).

`r_epsilon_psi` Corr(epsilon, psi).

`r_alpha_x` Corr(alpha, x).

`r_psi_x` Corr(psi, x).

`r_epsilon_x` Corr(epsilon, x).

`fix_total_var` Logical; if TRUE (default), residual variance is adjusted so total phenotypic variance is approx. 1.

Value

A list with:

- data: list of datasets
- params: named list of effect sizes
- design: sample design (n_ind, partners, repeats, iterations)

Examples

```
sim <- simulate_data(ind =1200, partners = 4, iterations = 100, B_0 = 1, Valpha=0.2, Vepsilon = 0.1)
```

summarise_results *Summarise bias and dispersion (MADm) across simulated fits*

Description

Summarise bias and dispersion (MADm) across simulated fits

Usage

```
summarise_results(results)
```

Arguments

`results`	Output from `run_model()`.

Value

Data frame with only parameters that were estimated in the Stan model.

Examples

```
if (requireNamespace("cmdstanr", quietly = TRUE) ||
    requireNamespace("rstan", quietly = TRUE)) {
  sim <- simulate_data(ind = 100, Valpha = 0.2, Vepsilon = 0.1, iterations = 2)
  res <- run_model(sim, model = "Trait.stan", iter = 500, cores = 2)
  summary(res)
} else {
  message("CmdStanR or rstan not available; example skipped.")
}
```

Index