Package: psynetsim (via r-universe)

May 16, 2026

Type Package

Title Simulating treatment intervention to symptom network

Version 0.2.0

Author Takumi Omizu

Maintainer Takumi Omizu `<tomizu@psy.senshu-u.ac.jp>`

Description More about what it does (maybe more than one line) Use four spaces when indenting paragraphs within the Description.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports cowplot, dplyr, ggplot2, grDevices, gridExtra, qgraph, scales, stats, tibble, tidyr, glmnet

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev

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

Date/Publication 2025-12-11 03:04:47 UTC

RemoteUrl https://github.com/TakumiOmizu/psynetsim

RemoteRef HEAD

RemoteSha cefb1291ca91e4b262efae4a9cb1b02a66513215

Contents

simulate_network_threshold 2
simulate_treatment_network 3
simulate_treatment_network_dym 5

Index **8**

`simulate_network_threshold` *Simulating Dynamic Treatment Intervention (Fixed X-Axis)*

Description

`simulate_network_threshold` simulates an intervention with duration control and fixed plot axis.

Usage

```
simulate_network_threshold(
  W_init,
  b_init,
  target,
  connectivity = 1,
  trial = 10,
  baseline_iteration = 10,
  intervention_duration = 20,
  num_updates = 5,
  update_interval = 10,
  follow_up_iteration = 10,
  symptom_name = NULL
)
```

Arguments

`W_init`	A square matrix representing the initial weighted connections.
`b_init`	A numeric vector representing the initial thresholds.
`target`	A numeric vector matching the number of symptoms. Values indicate the magnitude of intervention added to each symptom's threshold.
`connectivity`	A numeric value (default: 1) controlling overall connection strength.
`trial`	An integer (default: 10) specifying the number of simulation trials.
`baseline_iteration`	An integer (default: 10) steps before intervention.
`intervention_duration`	An integer (default: 20). How many steps the intervention lasts before being removed.
`num_updates`	An integer (default: 5). How many times the network should be re-estimated after intervention.
`update_interval`	An integer (default: 10). The number of simulation steps between each network re-estimation.
`follow_up_iteration`	An integer (default: 10) steps after the dynamic update phase.
`symptom_name`	A character vector for symptom names.

Value

A list containing plot_data, raw_data, final_network, network_history_full, result_plot, and result_text.

`simulate_treatment_network` *Simulating treatment intervention to symptom network*

Description

`simulate_treatment_network` simulates the effect of treatment interventions on a symptom network. This package, **ROXGAN**, provides tools for understanding and visualizing the dynamic impact of interventions within complex symptom interdependencies.

Usage

```
simulate_treatment_network(
  W_init,
  b_init,
  target,
  connectivity = 1,
  edge_between_TC = 1,
  weight_bias = 1,
  TB = 1,
  trial = 10,
  baseline_iteration = 10,
  num_TC = 5,
  TC_iteration_per_component = 10,
  follow_up_iteration = 10,
  symptom_name = NULL
)
```

Arguments

`W_init`	A square matrix representing the initial weighted connections (edges) between symptoms. The number of rows/columns determines the number of symptoms in the network.
`b_init`	A numeric vector representing the initial thresholds (biases) for each symptom. Its length must match the number of symptoms in `W_init`.
`target`	A list or vector indicating which symptoms are targeted by the treatment. Use 1 for targeted symptoms and 0 for non-targeted symptoms. Its length must match the number of symptoms.
`connectivity`	A numeric value (default: 1) controlling the overall strength of connections within the network. Higher values increase the influence of connected nodes.

`edge_between_TC`	A numeric value (default: 1) controlling the strength of connections between different treatment components (TCs).
`weight_bias`	A numeric value (default: 1) influencing the weight of connections between treatment components and targeted symptoms. A value greater than 1 increases the negative (inhibitory) effect, while less than 1 decreases it.
`TB`	A numeric value (default: 1) influencing the threshold (bias) of treatment components. Higher values make TCs less likely to activate spontaneously.
`trial`	An integer (default: 10) specifying the number of simulation trials to run. The final results are averaged across these trials.
`baseline_iteration`	An integer (default: 10) specifying the number of iterations for the baseline (pre-treatment) simulation phase.
`num_TC`	An integer (default: 5) specifying the total number of treatment components to be introduced during the simulation.
`TC_iteration_per_component`	An integer (default: 10) specifying the number of iterations for each treatment component introduction phase.
`follow_up_iteration`	An integer (default: 10) specifying the number of iterations for the follow-up (post-treatment) simulation phase.
`symptom_name`	A character vector (default: NULL) providing custom names for the symptoms. If NULL, symptoms will be named alphabetically (e.g., "a", "b", "c", ...). Its length must match the number of symptoms.

Value

A list containing:

`result_plot`	A `cowplot` object displaying two plots: a network visualization of the average weights (symptom and treatment components) and a time-series plot showing the mean and standard deviation of symptom and treatment component activation over the simulation period.
`result_text`	A character string summarizing the mean and standard deviation of symptom and treatment component activation at the final simulation step.

Examples

```
# Install and load necessary packages if not already installed
# install.packages(c("qgraph", "tidyverse", "cowplot", "gridExtra"))
library(qgraph)
library(dplyr)
library(tidyr)
library(tibble)
library(ggplot2)
library(cowplot)
library(gridExtra)
```

```
# Example data for a 6-symptom network
set.seed(456)
weight_6 <- matrix(rnorm(6*6, mean = 0.2, sd = 0.08), nrow = 6, ncol = 6)
diag(weight_6) <- 0
weight_6[upper.tri(weight_6)] <- t(weight_6)[upper.tri(weight_6)]
threshold_6 <- data.frame(threshold = rnorm(6, mean = 0.3, sd = 0.05))
target_list_6 <- list(symptom1 = 1, symptom2 = 0, symptom3 = 1,
                      symptom4 = 0, symptom5 = 0, symptom6 = 1)
custom_symptom_names_6 <- c("Anxiety", "Sadness", "Fatigue",
                            "Insomnia", "Irritability", "Pain")

# Run the simulation with custom parameters
sim_results <- simulate_treatment_network(
  W_init = weight_6,
  b_init = threshold_6$threshold,
  target = target_list_6,
  connectivity = 1.2,
  edge_between_TC = 0.8,
  weight_bias = 1.2,
  TB = 0.8,
  trial = 5, # Example: Overriding default 10
  baseline_iteration = 15, # Example: Overriding default 10
  num_TC = 4, # Example: Overriding default 5
  TC_iteration_per_component = 12, # Example: Overriding default 10
  follow_up_iteration = 15, # Example: Overriding default 10
  symptom_name = custom_symptom_names_6
)

# Print summary text and display plots
print(sim_results$result_text)
print(sim_results$result_plot)

# Run the simulation with only required parameters (using all default iteration/trial values)
sim_results_default <- simulate_treatment_network(
  W_init = weight_6,
  b_init = threshold_6$threshold,
  target = target_list_6
)
print(sim_results_default$result_text)
print(sim_results_default$result_plot)
```

`simulate_treatment_network_dym`

Simulating Dynamic Treatment Intervention on Symptom Network

Description

`simulate_treatment_network_dym` simulates treatment intervention on a symptom network, where **symptom network connections dynamically update** based on recent patient state data using regularized logistic regression (Lasso/Elastic Net), capturing how the system responds to treatment over time. This functionality is an extension of the base `simulate_treatment_network` function.

Usage

```
simulate_treatment_network_dym(
  W_init,
  b_init,
  target,
  connectivity = 1,
  edge_between_TC = 1,
  weight_bias = 1,
  TB = 1,
  trial = 10,
  baseline_iteration = 10,
  num_TC = 5,
  TC_iteration_per_component = 10,
  follow_up_iteration = 10,
  symptom_name = NULL
)
```

Arguments

`W_init`	A square matrix representing the initial weighted connections (edges) between symptoms. The number of rows/columns determines the number of symptoms in the network.
`b_init`	A numeric vector representing the initial thresholds (biases) for each symptom. Its length must match the number of symptoms in `W_init`.
`target`	A list or vector indicating which symptoms are targeted by the treatment. Use 1 for targeted symptoms and 0 for non-targeted symptoms. Its length must match the number of symptoms.
`connectivity`	A numeric value (default: 1) controlling the overall strength of connections within the network. Higher values increase the influence of connected nodes.
`edge_between_TC`	A numeric value (default: 1) controlling the strength of connections between different treatment components (TCs).
`weight_bias`	A numeric value (default: 1) influencing the weight of connections between treatment components and targeted symptoms. A value greater than 1 increases the negative (inhibitory) effect.
`TB`	A numeric value (default: 1) influencing the threshold (bias) of treatment components. Higher values make TCs less likely to activate spontaneously.
`trial`	An integer (default: 10) specifying the number of simulation trials to run. Results are averaged across trials.
`baseline_iteration`	An integer (default: 10) specifying the number of iterations for the baseline (pre-treatment) simulation phase.

`num_TC`	An integer (default: 5) specifying the total number of treatment components to be introduced.
`TC_iteration_per_component`	An integer (default: 10) specifying the number of iterations after each TC introduction for the network to adapt, before re-estimation.
`follow_up_iteration`	An integer (default: 10) specifying the number of iterations for the follow-up phase.
`symptom_name`	A character vector (default: NULL) providing custom names for the symptoms. If NULL, symptoms will be named alphabetically.

Value

A list containing:

`result_plot`	A `cowplot` object displaying two plots: a network visualization of the *initial* symptom connections (which are dynamically updated during simulation) and a time-series plot showing the mean and standard deviation of symptom and treatment component activation.
`result_text`	A character string summarizing the mean and standard deviation of symptom and treatment component activation at the final simulation step.

Index

simulate_network_threshold, 2
simulate_treatment_network, 3
simulate_treatment_network_dym, 5