Package: netmatchRI (via r-universe)

October 16, 2026

Title Network-Constrained Matching and Randomization-Based Inference

Version 0.1.0

Author Zhejia Dong [aut, cre]

Maintainer Zhejia Dong <zhejia_dong@brown.edu>

Description Provides network-constrained matching and randomization-based inference for observational network data with dependence and interference. Dual-penalty matching minimizes total covariate dissimilarity while enforcing penalties on network proximity between and within treatment arms in matched sets. Randomization-based inference combines evidence from correlated matched sets based on this design. The package also provides covariate-balance and network-distance diagnostics, sensitivity analysis, and simulated example data. The open-source 'HiGHS' solver is used by default to solve the matching problem. An optional interface to the commercial 'Gurobi Optimizer' is available. Eligible academic users can request a free license at <https://www.gurobi.com/academics/>. The optimizer can be downloaded from <https://www.gurobi.com/product/download-center>. The associated 'gurobi' R package is not available from CRAN; it is distributed with the optimizer and can be installed by following the official instructions at <https://docs.gurobi.com/projects/optimizer/en/current/reference/r/setup.html>.

License MIT + file LICENSE

URL https://github.com/zhejiadong/netmatchRI

BugReports https://github.com/zhejiadong/netmatchRI/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports ggplot2, graphics, highs, igraph, Matrix, mvtnorm, optmatch, Rglpk, slam, stats, utils

Suggests knitr, MASS, rmarkdown, testthat (>= 3.2.0)

Enhances gurobi

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake libglpk-dev libxml2-dev pkg-config

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

Date/Publication 2026-10-16 01:03:55 UTC

RemoteUrl https://github.com/zhejiadong/netmatchri

RemoteRef HEAD

RemoteSha 36adca93b353493bda9109977522ba9e6d274d0b

Contents

netmatchRI-package . . . 2
critical_sensitivity . . . 3
diagnose_match . . . 4
matched_data . . . 5
netmatch . . . 6
netmatch_example . . . 7
plot_covariate_balance . . . 8
plot_covariate_similarity . . . 9
plot_sensitivity . . . 9
RI_adjusted . . . 10
RI_design . . . 11
RI_unadjusted . . . 12
sensitivity_grid . . . 13
simulate_netmatch_example . . . 14

Index **16**

`netmatchRI-package` *netmatchRI: Network-Constrained Matching and Randomization-Based Inference*

Description

netmatchRI provides network-constrained matching and randomization-based inference for observational network data. Dual-penalty matching minimizes total covariate dissimilarity while enforcing penalties on network proximity between and within treatment arms in matched sets. Randomization-based inference combines evidence from correlated matched sets based on this design.

Details

Use `netmatch()` to construct a dual-penalty, single-penalty, or covariate-only matched design. Inspect the result with `summary()`, `matched_data()`, and `diagnose_match()`, which checks covariate balance and within-set network distances. Compare selected matching designs with `plot_covariate_balance()` and `plot_covariate_similarity()`. Use `RI_unadjusted()`, `RI_adjusted()`, or `RI_design()` for inference, and use `sensitivity_grid()`, `critical_sensitivity()`, and `plot_sensitivity()` to examine sensitivity to residual network dependence.

Dual-penalty matching uses the open-source HiGHS solver by default. The `solver` argument can select HiGHS, Gurobi, or GLPK; `solver = "auto"` tries them in that order of availability.

`netmatch_example` contains the fixed data and network used in the guide. `simulate_netmatch_example()` generates new simulated network data.

`critical_sensitivity` *Critical Sensitivity Curve and Critical Ratio*

Description

Computes critical eta values where the p-value equals `alpha`, together with critical eta at `rho = 1` and the critical ratio.

Usage

```
critical_sensitivity(match, outcome, rho = seq(0, 1, by = 0.01),
  alpha = 0.05, kappa = NULL, weight_type = c("ns", "ntc"))
```

Arguments

`match`	A `netmatch` object. Critical-curve interpretation is intended for dual-penalty matching.
`outcome`	Name of the numeric outcome column.
`rho`	Sensitivity-decay values in `[0, 1]`. Default: `seq(0, 1, by = 0.01)`.
`alpha`	Test level. Default: `0.05`.
`kappa`	Network distance threshold. `NULL` uses `match$kappa`.
`weight_type`	Matched-set weighting. `"ns"` uses $1/(n_s + 1)$, where n_s is the set size; `"ntc"` uses $1/(n_{st} n_{sc})$, where n_{st} and n_{sc} are the treated and control counts. Default: `"ns"`.

Details

Let $h_{s,l}$ be the minimum network distance between units in matched sets s and l. For pairs with $h_{s,l} \leq \kappa$, the sensitivity bound is $M^*_{sl}(\eta, \rho) = \eta \rho^{h_{s,l}-1} M_{sl}$. At a critical contour point, its value is the critical sensitivity bound $M^{\text{crit}}_{sl}(\rho) = M^*_{sl}(\eta^*(\rho), \rho)$. Within the parameter range, eta values below the critical curve retain rejection and values above it do not. On the curve, the p-value equals `alpha`.

Let S be the number of matched sets and N_κ the number of relevant matched-set pairs. The critical ratio is

$$R_{\text{crit}} = \frac{N_\kappa^{-1} \sum_{s<l:\, l \in \mathcal{I}_\kappa(s)} w_s w_l M_{sl}^{\text{crit}}(\rho)}{S^{-1} \sum_{s=1}^{S} w_s^2 \sigma_{ss}}.$$

Here, w_s is the matched-set weight and σ_{ss} is the variance of its Mann–Whitney U statistic. The numerator averages the weighted covariance bounds over pairs with minimum distance at most `kappa`; the denominator averages the weighted within-set variances. The ratio is constant along the critical curve.

The summary's `critical_ratio` is `NA` when no critical contour is attained within the range. If the baseline p-value is greater than `alpha`, critical eta is `NA`. If the baseline p-value equals `alpha` and the sensitivity denominator is positive, critical eta and the ratio are zero.

Value

A `netmatch_critical_sensitivity` object. `summary` reports the critical eta at `rho = 1` and the critical ratio; `curve` contains the critical eta over the requested rho values; `detail` contains the supporting counts and averages. The object also retains the analysis settings and matched design.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2)
crit <- critical_sensitivity(m, "Y", rho = seq(0, 1, by = 0.25))
crit
crit$summary
crit$curve
```

`diagnose_match` *Diagnose a Matched Design*

Description

Checks covariate balance and network distance within matched sets.

Usage

```
diagnose_match(match)
```

Arguments

`match`	A `netmatch` object.

Details

Each factor level is represented by an indicator. For every reported covariate, the before and after absolute standardized mean differences use one fixed pooled original-sample standard deviation,

$$\sqrt{\{(n_1 - 1)s_1^2 + (n_0 - 1)s_0^2\}/(n_1 + n_0 - 2)}.$$

The before difference uses unweighted original-sample treatment-group means. The after difference uses the unit matching weights among matched units.

Value

A list containing covariate-balance results, a network-distance summary, and the within-matched-set distance table.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
diag <- diagnose_match(m)
diag$covariate_balance
diag$network_summary
diag$within_distance_table
```

---

matched_data *Extract Data from a Matched Design*

Description

Returns the original unit-level data with matched-set membership and matching weights.

Usage

```
matched_data(object, drop_unmatched = TRUE)
```

Arguments

`object`	A `netmatch` object.
`drop_unmatched`	If `TRUE`, omit unmatched units. Default: `TRUE`.

Value

The unit-level data with `subclass` and `weights`.

netmatch *Build a Network-Constrained Matched Design*

Description

Provides three matching methods for observational network data: dual-penalty, single-penalty, and covariate-only matching. Dual-penalty matching minimizes total covariate dissimilarity while restricting network-close pairs both between and within treatment arms in matched sets. Single-penalty matching restricts network-close pairs across treatment arms; covariate-only matching has no network-distance restriction.

Usage

```
netmatch(data, treat, covariates, network,
  method = c("dual", "single", "covariate"), kappa = 2,
  solver = c("highs", "auto", "gurobi", "glpk"), min_controls = 0.01,
  max_controls = 100, caliper = Inf, timelimit = 90, mipgap = 0.01,
  threads = 1, estimand = c("ATT", "ATC", "ATE"),
  network_type = c("auto", "adjacency", "distance"),
  include_solver = FALSE)
```

Arguments

`data`	Data frame with one row per unit. The names `subclass` and `weights` are reserved.
`treat`	Name of the binary treatment column.
`covariates`	Names of the numeric or factor covariates used for matching.
`network`	Symmetric adjacency or network-distance matrix aligned with `data`.
`method`	Matching method: dual-penalty, single-penalty, or covariate-only. Default: dual-penalty (`"dual"`).
`kappa`	Network distance threshold used by designs with network restrictions. Default: `2`.
`solver`	Solver for dual-penalty matching. `"auto"` tries HiGHS, then Gurobi, then GLPK. Default: `"highs"`.
`min_controls`	Minimum control-to-treated ratio. Default: `0.01`.
`max_controls`	Maximum control-to-treated ratio; use `Inf` for no upper limit. Default: `100`.
`caliper`	Mahalanobis-distance caliper for treated-control pairs in dual-penalty matching, analogous to a propensity-score caliper; use `Inf` for no caliper. Default: `Inf`.
`timelimit`	Solver time limit in seconds. Default: `90`.
`mipgap`	Target relative MIP gap. Default: `0.01`.
`threads`	Requested solver threads. Default: `1`.

`estimand`	Unit matching weights used in summaries and balance diagnostics: `"ATT"`, `"ATC"`, or `"ATE"`. Default: `"ATT"`. This choice does not change the sharp-null RI test.
`network_type`	Format of `network`: `"auto"`, `"adjacency"`, or `"distance"`. Default: `"auto"`.
`include_solver`	If `TRUE`, retain the complete solver result. Default: `FALSE`.

Details

Dual-penalty matching requires every pair of units in a matched set to have network distance greater than `kappa`. Single-penalty matching imposes this restriction only on treated-control pairs.

Covariate dissimilarity is the square root of the Mahalanobis quadratic form, using the pooled within-treatment-group covariance matrix.

Dual-penalty matching is formulated and solved as a mixed-integer program. Covariate-only and single-penalty comparison designs use `optmatch::fullmatch()`.

Value

`netmatch()` returns a `netmatch` object containing the matched data, full-sample subclasses and unit weights, design settings, network distances, and solver summary. The complete solver result is included only when requested.

`summary()` returns sample, balance, network, and solver summaries. `plot()` draws the within-matched-set matching diagnostic.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(
  data = sim$data,
  treat = "Z",
  covariates = c("X1", "X2", "X3"),
  network = sim$net_dist,
  method = "dual",
  kappa = 2,
  solver = "highs",
  estimand = "ATT"
)
m
summary(m)
```

`netmatch_example` *Simulated Network Data for the Analysis Workflow*

Description

Simulated data with no treatment effect and network-dependent covariates, treatment assignment, and outcomes. The dataset was selected to illustrate how sensitivity to network dependence can change the inferential conclusion.

Usage

```
data(netmatch_example)
```

Format

A list containing `data`, a data frame with 300 rows and columns `Z`, `X1`, `X2`, `X3`, and `Y`; `Adj`, the aligned 300 by 300 undirected adjacency matrix; and `net_dist`, the corresponding shortest-path distances (infinite between disconnected nodes). Rows of the matrices follow the data frame's row order.

Source

Generated with `simulate_netmatch_example(seed = 17, n = 300, beta_z = 0, alpha1 = 0.9, alpha2 = 0.3, edge_prob = 4 / 299)`. The repository data-generation script records the construction.

Examples

```
data(netmatch_example)
dim(netmatch_example$data)
```

---

plot_covariate_balance *Plot Covariate Balance*

Description

Compares absolute standardized mean differences before matching and across selected matching designs.

Usage

```
plot_covariate_balance(designs, include_before = TRUE)
```

Arguments

`designs`	Named list of comparable `netmatch` objects.
`include_before`	Include the unmatched sample. Default: `TRUE`.

Value

A `ggplot` object.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2)
plot_covariate_balance(list("Dual-penalty" = m))
```

plot_covariate_similarity *Plot Covariate Similarity*

Description

Compares average treated-control Mahalanobis distance before matching and within matched sets across selected matching designs. Smaller values indicate greater covariate similarity.

Usage

```
plot_covariate_similarity(designs, include_before = TRUE)
```

Arguments

`designs`	Named list of comparable `netmatch` objects.
`include_before`	Include the unmatched sample. Default: `TRUE`.

Value

A `ggplot` object.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2)
plot_covariate_similarity(list("Dual-penalty" = m))
```

plot_sensitivity *Plot Sensitivity Results*

Description

Plots a p-value grid or a critical sensitivity curve. Stored analysis choices, including the weighting method, are reused for conversions and unadjusted reference lines.

Usage

```
plot_sensitivity(x, type = NULL, alpha = 0.05, unadjusted = NULL,
  critical_ylim = NULL)
```

Arguments

x	A sensitivity-grid or critical-sensitivity object.
type	"pvalue" or "critical". The default follows the class of x.
alpha	Reference or test level. Default: 0.05.
unadjusted	Optional unadjusted p-value or test result for the p-value plot.
critical_ylim	Optional y-axis limits for the critical plot. Default: NULL, which scales to the curve within [0, 1].

Details

A p-value plot shows how the p-value changes with rho for each selected value of eta. A critical plot shows the boundary value of eta; values below the curve retain rejection at the selected alpha, and values above it do not. If no finite boundary exists on the requested grid, the function returns an error with the reason rather than drawing a curve. Algebraic crossings outside the displayed range remain in the plot data but are not critical contour points within the parameter range.

Value

A ggplot object.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
sens <- sensitivity_grid(m, "Y", eta = c(0, 0.01, 0.05, 0.10, 0.15, 0.20),
                         rho = seq(0, 0.5, by = 0.05))
plot_sensitivity(sens)

crit <- critical_sensitivity(m, "Y")
plot_sensitivity(crit)
```

RI_adjusted *Randomization-Based Inference Adjusting for Covariance*

Description

Tests the sharp null that no unit's outcome is affected by any unit's treatment, including both direct effects and treatment interference. The adjusted test scales the design-based covariance bounds between matched sets using eta and rho.

Usage

```
RI_adjusted(match, outcome, eta = 0.03, rho = 0.10, kappa = NULL,
  weight_type = c("ns", "ntc"))
```

Arguments

`match`	A `netmatch` object.
`outcome`	Name of the numeric outcome column.
`eta`	Relative magnitude of the design-based covariance bound in `[0, 1]`. Default: `0.03`.
`rho`	Decay parameter in `[0, 1]`. Default: `0.10`.
`kappa`	Network distance threshold. `NULL` uses `match$kappa`.
`weight_type`	Matched-set weighting. `"ns"` uses $1/(n_s + 1)$, where n_s is the set size; `"ntc"` uses $1/(n_{st} n_{sc})$, where n_{st} and n_{sc} are the treated and control counts. Default: `"ns"`.

Details

The test uses a weighted sum of Mann-Whitney U statistics from matched sets and a normal approximation to obtain a two-sided p-value. The design-based covariance bound between matched sets is scaled by `eta` and decays with the minimum network distance between sets according to `rho`. Smaller `rho` gives faster decay, and `rho = 1` gives no decay. These are sensitivity parameters, not estimated covariances. The adjusted bound must cover the cross-matched-set covariance for the conservative inference described in the paper to apply. The null moments assume no within-matched-set ties.

Value

A `netmatch_test` object containing the test result, matched-set details, covariance and distance matrices, analysis settings, and original matched design.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
RI_adjusted(m, "Y", eta = 0.05, rho = 0.20)
```

RI_design — *Randomization-Based Inference with Design-Based Covariance Bounds*

Description

Tests the sharp null that no unit's outcome is affected by any unit's treatment, including both direct effects and treatment interference. The design-based test accounts for dependence using covariance bounds between matched sets.

Usage

```
RI_design(match, outcome, kappa = NULL, weight_type = c("ns", "ntc"))
```

Arguments

`match`	A `netmatch` object.
`outcome`	Name of the numeric outcome column.
`kappa`	Network distance threshold. `NULL` uses `match$kappa`.
`weight_type`	Matched-set weighting. `"ns"` uses $1/(n_s + 1)$, where n_s is the set size; `"ntc"` uses $1/(n_{st} n_{sc})$, where n_{st} and n_{sc} are the treated and control counts. Default: `"ns"`.

Details

The test uses a weighted sum of Mann-Whitney U statistics from matched sets and a normal approximation to obtain a two-sided p-value. It uses the design-based covariance bounds without scaling or decay for pairs of matched sets whose minimum network distance is at most `kappa`. The null moments assume no within-matched-set ties.

Value

A `netmatch_test` object containing the test result, matched-set details, covariance and distance matrices, analysis settings, and original matched design.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
RI_design(m, "Y")
```

`RI_unadjusted` *Randomization-Based Inference Without Covariance Adjustment*

Description

Tests the sharp null that no unit's outcome is affected by any unit's treatment, including both direct effects and treatment interference. The unadjusted test proceeds without adjusting for covariance between matched sets.

Usage

```
RI_unadjusted(match, outcome, kappa = NULL, weight_type = c("ns", "ntc"))
```

Arguments

`match`	A `netmatch` object.
`outcome`	Name of the numeric outcome column.
`kappa`	Network distance threshold. `NULL` uses `match$kappa`.
`weight_type`	Matched-set weighting. `"ns"` uses $1/(n_s + 1)$, where n_s is the set size; `"ntc"` uses $1/(n_{st} n_{sc})$, where n_{st} and n_{sc} are the treated and control counts. Default: `"ns"`.

Details

The test uses a weighted sum of Mann-Whitney U statistics from matched sets and a normal approximation to obtain a two-sided p-value. It uses only within-matched-set variances. The null moments assume no within-matched-set ties.

Value

A `netmatch_test` object containing the test result, matched-set details, covariance and distance matrices, analysis settings, and original matched design.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
RI_unadjusted(m, "Y")
```

`sensitivity_grid` *Sensitivity Grid for Inferential Results*

Description

Evaluates how the p-value changes over a grid of eta and rho values. Eta scales the design-based covariance bound between matched sets. Rho controls how that bound decays with the minimum network distance between sets: for distance `d`, the multiplier is `eta * rho^(d - 1)`.

Usage

```
sensitivity_grid(match, outcome, eta = seq(0, 0.10, by = 0.01),
  rho = seq(0, 0.50, by = 0.05), kappa = NULL,
  weight_type = c("ns", "ntc"))
```

Arguments

`match`	A `netmatch` object.
`outcome`	Name of the numeric outcome column.
`eta`	Sensitivity-scale values in `[0, 1]`. Default: `seq(0, 0.10, by = 0.01)`.
`rho`	Sensitivity-decay values in `[0, 1]`. Default: `seq(0, 0.50, by = 0.05)`.
`kappa`	Network distance threshold. `NULL` uses `match$kappa`.
`weight_type`	Matched-set weighting. `"ns"` uses $1/(n_s + 1)$, where n_s is the set size; `"ntc"` uses $1/(n_{st} n_{sc})$, where n_{st} and n_{sc} are the treated and control counts. Default: `"ns"`.

Details

The set statistic counts a treated outcome only when it is strictly greater than a control outcome. The null moments assume no tied outcomes within a matched set.

Value

A `netmatch_sensitivity` object containing the p-value grid, original matched design, and analysis settings.

Examples

```
data("netmatch_example", package = "netmatchRI")
sim <- netmatch_example
m <- netmatch(sim$data, "Z", c("X1", "X2", "X3"), sim$net_dist,
              method = "dual", kappa = 2, solver = "auto")
sens <- sensitivity_grid(m, "Y", eta = c(0, 0.01, 0.05, 0.10, 0.15, 0.20),
                         rho = seq(0, 0.5, by = 0.05))
sens$grid
```

`simulate_netmatch_example` *Generate Example Network Data*

Description

Generates a random network and a data frame with three covariates, a binary treatment, and an outcome. Covariates, treatment, and outcome may be correlated across connected units.

Usage

```
simulate_netmatch_example(seed = NULL, beta_z = 0, n = 200,
  alpha1 = 0.9, alpha2 = 0.1, edge_prob = 0.02)
```

Arguments

`seed`	Simulation seed. Default: `NULL`, which uses the current RNG state.
`beta_z`	Treatment effect in the outcome model. Default: `0`.
`n`	Number of units. Default: `200`.
`alpha1`	Network-related covariance coefficient in `[0, 1]`. Default: `0.9`.
`alpha2`	Unit-specific covariance coefficient in `[0, 1]`. Default: `0.1`.
`edge_prob`	Edge probability. Default: `0.02`.

Value

A list containing the simulated `data`, adjacency matrix `Adj`, network-distance matrix `net_dist`, covariance matrix `V`, and the simulation settings.

Examples

```
sim <- simulate_netmatch_example(seed = 2, n = 200, beta_z = 0,
                                 alpha1 = 0.9, alpha2 = 0.1, edge_prob = 0.02)
dim(sim$data)
table(sim$data$Z)
```

Index

∗ **datasets**
    netmatch_example, 7

critical_sensitivity, 3, *3*

diagnose_match, *3*, 4

matched_data, *3*, 5

netmatch, *3*, 6
netmatch_example, *3*, 7
netmatchRI *(*netmatchRI-package*)*, 2
netmatchRI-package, 2

plot_covariate_balance, *3*, 8
plot_covariate_similarity, *3*, 9
plot_sensitivity, *3*, 9

RI_adjusted, *3*, 10
RI_design, *3*, 11
RI_unadjusted, *3*, 12

sensitivity_grid, *3*, 13
simulate_netmatch_example, *3*, 14