

Package ‘scNPM’

October 14, 2026

Type Package

Title Single-Cell Network Propensity Matching

Version 1.0.6

Date 2026-10-02

Description Estimates and compares heterogeneous gene conditional-dependence networks from single-cell expression data. A latent Gaussian mixture and probit observation mechanism represent cellular heterogeneity and zero observations. Protein interaction information supplies edge-specific prior weights, while network propensity matching links cell clusters across conditions and guides adaptation. Staged inference freezes stable matching, disables feedback, and retains baseline-model samples for posterior group and directional differential edge selection. Functions assess balance, record criterion diagnostics, and visualize within-condition and differential networks.

License GPL (>= 3)

Encoding UTF-8

LazyData true

LazyDataCompression xz

Imports Rcpp (>= 1.0.10), Matrix, igraph (>= 1.3.0), ggplot2 (>= 3.3.0), ggrepel (>= 0.9.0), scales (>= 1.2.0), stats, utils, tools

LinkingTo Rcpp, RcppArmadillo (>= 0.12.6.4.0), RcppParallel (>= 5.1.7)

SystemRequirements C++17, OpenMP

Depends R (>= 4.0.0)

URL https://github.com/taostat1/scNPM_R_package

BugReports https://github.com/taostat1/scNPM_R_package/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Tao Liu [aut, cre]
Maintainer Tao Liu <25210690190@m.fudan.edu.cn>
Repository CRAN
Date/Publication 2026-10-14 11:20:02 UTC

Contents

cell	2
cols_to_MR	3
cols_to_PR	3
compute_qf_smd	4
goal_gene	5
goal_gene_mt	5
mt	6
plot_diff_network	7
plot_group_network	8
run_scNPM	9
select_differential_edges	15
select_group_edges	16
Index	18

cell	<i>Cell-Level Metadata (AMD Example Data)</i>
------	---

Description

A data frame with per-cell metadata (quality-control metrics computed with **scater**, sample provenance and the PR_or_MR region indicator) for the 400 cells of the AMD example subset (200 PR, 200 MR). Rows are aligned with the columns of goal_gene_mt. The first example covariate column in [run_scNPM](#) is typically built from PR_or_MR (1 = MR, treatment; 0 = PR, control).

Usage

data(top50_goal_gene_mt)

Format

A data frame with 400 rows and 52 columns: cell barcodes (Barcode), cell type labels (Labels), the region indicator PR_or_MR, donor identifier (individual), tissue, spatial coordinates (x, y), and **scater** quality-control metrics (columns prefixed scater_qc.: library size, detected features, percentage of counts in top features, mitochondrial percentage, etc.).

Source

Derived from a public single-cell RNA-seq study of human retinas (see the scNPM reference publication).

cols_to_MR*Column Indices of MR Cells (AMD Example Data)*

Description

Integer vector of column indices (in goal_gene_mt/mt) of the cells belonging to the MR group, used as the treatment arm in the example analysis.

Usage

```
data(top50_goal_gene_mt)
```

Format

An integer vector of length 200.

Source

Derived from the PR_or_MR metadata of the example dataset.

cols_to_PR*Column Indices of PR Cells (AMD Example Data)*

Description

Integer vector of column indices (in goal_gene_mt/mt) of the cells belonging to the PR group, used as the control arm in the example analysis.

Usage

```
data(top50_goal_gene_mt)
```

Format

An integer vector of length 200.

Source

Derived from the PR_or_MR metadata of the example dataset.

compute_qf_smd	<i>Compute the SMD of the Network Quadratic Form for Matched Cluster Pairs</i>
----------------	--

Description

In all-reference mode (version 1.0.3), this helper returns one row per matched pair and reference network, using the exact final network features. The additional column ‘network_feature’ identifies each reference. The legacy vector interface below remains supported for older results.

Usage

```
compute_qf_smd(result_list)
```

Arguments

`result_list` Output of [run_scNPM()] (as saved with [save()]).

Details

The legacy network quadratic form $qf_i = \theta_i' \Omega_{k(i)} \theta_i$ enters the propensity score model as an additional covariate. This helper reports the matched-pair SMD of the quadratic form using the same formula as ‘smd_covariate_results’:

$$SMD = |mean_t - mean_c| / \sqrt{(var_t + var_c)/2}$$

Two modes are supported (returned in the attribute ‘mode’):

- **exact**: ‘result_list’ contains ‘qf_global’ and ‘final_group_label’ (scNPM >= 1.0.1). The helper uses the quadratic forms and labels from the internal final matching call.
- **posterior-approx**: for results saved by older versions (with the heavy ‘Result’ element stripped), the quadratic form is recomputed from stored sampling summaries (‘theta_est’, ‘invcov_treat_est’, ‘invcov_ctrl_est’, ‘group_est’). The legacy mode name is retained for compatibility; these summaries need not approximate the final matching state closely.

Value

A data frame with columns ‘treat_cluster’, ‘matched_control_cluster’ and ‘qf_smd’, rounded to four decimal places. All-reference results additionally include ‘network_feature’ and contain $K * (2 * K)$ rows; their mode attribute is ‘exact-all-reference’. Unmatched or empty pairs have NA. Singleton samples have variance zero. At the numerical zero-variance boundary, equal means give zero and unequal means give infinity, using the matching routine’s tolerances.

Examples

```
# Small synthetic example (exact mode), runs in well under 5 seconds
set.seed(1)
G <- 5; K <- 2; nT <- 30; nC <- 30
invT <- array(diag(G), c(G, G, K)); invC <- array(diag(G), c(G, G, K))
thT <- matrix(rnorm(nT * G), nT, G)
thC <- matrix(rnorm(nC * G, mean = 0.5), nC, G)
qfT <- rowSums((thT %>% invT[, , 1]) * thT)
qfC <- rowSums((thC %>% invC[, , 2]) * thC)
r1 <- list(
  K = K,
  matching_result_df = data.frame(treatment_cluster = 1:K,
                                   matched_control_cluster = c(2, 1)),
  qf_global = c(qfT, qfC),
  final_group_label = c(rep(0, nT), rep(K + 1, nC))
)
compute_qf_smd(r1)
```

goal_gene	<i>Target Gene Names (AMD Example Data)</i>
-----------	---

Description

Character vector with the symbols of the 50 target genes (rows of goal_gene_mt).

Usage

```
data(top50_goal_gene_mt)
```

Format

A character vector of length 50 (e.g., "FTH1", "SAG", "TF", "CLU", "PDE6G", "GLUL", ...).

Source

Selected in the scNPM reference publication.

goal_gene_mt	<i>Top-50 Gene Expression Matrix (AMD Example Data)</i>
--------------	---

Description

Sparse UMI count matrix (genes x cells) for the 50 target genes used in the AMD (age-related macular degeneration) example analysis. This is a stratified random subset (200 PR control and 200 MR treatment cells) of the full 20091-cell dataset, drawn with a fixed seed. Loaded via data(top50_goal_gene_mt) together with the companion objects cell, goal_gene, mt, cols_to_PR and cols_to_MR.

Usage

```
data(top50_goal_gene_mt)
```

Format

A [dgTMatrix](#) with 50 rows (genes) and 400 columns (cells); entries are raw UMI counts.

Source

Derived from a public single-cell RNA-seq study of human retinas (see the scNPM reference publication).

mt

Full Gene Expression Matrix (AMD Example Data)

Description

Sparse UMI count matrix (all detected genes x cells) from which the top-50 target gene matrix `goal_gene_mt` was extracted. Provided so users can derive alternative gene subsets. This object covers the same 400 cells as `goal_gene_mt`; genes detected in fewer than 3 of these cells were removed.

Usage

```
data(top50_goal_gene_mt)
```

Format

A [dgTMatrix](#) with 12530 rows (genes) and 400 columns (cells); entries are raw UMI counts.

Source

Derived from a public single-cell RNA-seq study of human retinas (see the scNPM reference publication).

plot_diff_network	<i>Plot Differential Gene Regulatory Network (Combined/Separate Mode)</i>
-------------------	---

Description

Plots differential networks (treatment - control) with flexible output: - Combined mode: solid/dashed edges in a single plot (default) - Separate mode: solid/dashed edges as two independent plots

Usage

```
plot_diff_network(
  diff_mat,
  gene_names,
  solid_threshold = 0.5,
  dashed_threshold = -0.5,
  solid_color = "#ADD8E6",
  dashed_color = "#ADD8E6",
  output_mode = c("combined", "separate"),
  ...
)
```

Arguments

diff_mat	Differential matrix (treatment - control, rows/columns = gene names). Matrices returned by Bayesian-mode run_scNPM carry a scNPM_selection attribute; its selected edge masks are used directly, without re-thresholding.
gene_names	Vector of gene names (matches diff_mat row/column names).
solid_threshold	Minimum positive difference for unannotated matrices (default = 0.5); ignored for matrices carrying Bayesian selection masks.
dashed_threshold	Maximum negative difference for unannotated matrices (default = -0.5); ignored for matrices carrying Bayesian selection masks.
solid_color	Color for solid edges (treatment-enhanced, default = "#ADD8E6", the original scNPM light blue; set e.g. "#2ECC71" for green).
dashed_color	Color for dashed edges (treatment-reduced, default = "#ADD8E6", the original scNPM light blue; set e.g. "#F39C12" for orange).
output_mode	Output type: "combined" (single plot, default) or "separate" (two plots).
...	Additional parameters passed to plot_network_core() (separate mode) or internal plotting (combined mode).

Value

- If output_mode = "combined": ggplot object (single plot with solid/dashed edges) or NULL (no edges) - If output_mode = "separate": list of ggplot objects (solid/dashed) or NULL for empty edges

Examples

```
# Self-contained example with a small random differential matrix (< 5 sec)
set.seed(1)
gn <- paste0("Gene_", 1:10)
dm <- matrix(rnorm(100), 10, 10, dimnames = list(gn, gn))
dm <- (dm + t(dm)) / 2
diag(dm) <- 0

# Mode 1 - Combined (solid+dashed in one plot, default colors: light blue)
p_combined <- plot_diff_network(
  diff_mat = dm,
  gene_names = gn,
  output_mode = "combined"
)

# Mode 2 - Separate (solid/dashed as two plots; custom colors optional)
p_separate <- plot_diff_network(
  diff_mat = dm,
  gene_names = gn,
  output_mode = "separate",
  solid_color = "#2ECC71",
  dashed_color = "#F39C12",
  label_size = 3
)
```

plot_group_network	<i>Plot a Selected Group Network</i>
--------------------	--------------------------------------

Description

Uses the saved BFDR mask from [select_group_edges()] or a staged [run_scNPM()] fit. No probability rethresholding occurs in BFDR mode.

Usage

```
plot_group_network(
  edge_mat,
  gene_names = NULL,
  group_type = c("treatment", "control"),
  edge_color = "#ADD8E6",
  ...,
  selection = c("bfdr", "frequency"),
  frequency_threshold = 0.5
)
```

Arguments

edge_mat	A named network entry such as 'fit\$group_fdr\$networks\$treatment_1', or its annotated 'probability_matrix'. An unannotated frequency matrix is accepted only in frequency mode.
----------	---

gene_names	Gene names in the same order as the matrix. Omit to use the row names stored in the BFDR network.
group_type	Treatment or control; used for an empty-network message.
edge_color	Network edge color.
...	Additional parameters passed to plot_network_core().
selection	'bfdr' (default) uses the saved decision mask. 'frequency' explicitly requests the former descriptive display rule.
frequency_threshold	Cutoff for frequency mode only, default 0.5 with the inclusive >= operator. It is not a BFDR rule.

Value

A ggplot object or NULL for an empty selected network.

Examples

```
# Synthetic retained posterior summaries for one matched pair.
# These values illustrate selection; no MCMC is run in this example.
gn <- paste0("G", 1:3)
p <- matrix(c(0,.99,.97,.99,0,.60,.97,.60,0), 3,
            dimnames = list(gn, gn))
z <- array(0, c(3,3,1))
p10 <- array(p, c(3,3,1)); p00 <- array(1-p, c(3,3,1))
diag(p00[, , 1]) <- 0
example_fit <- list(
  K = 1L, G = 3L, gene_names = gn, inference_mode = "staged",
  matching_result_df = data.frame(treatment_cluster=1L, matched_control_cluster=1L),
  sampling_schedule = list(feedback_disabled_for_retained=TRUE, retained_draws=500L),
  joint_edge_probabilities = list(p00=p00, p10=p10, p01=z, p11=z)
)
group <- select_group_edges(example_fit)
plot_group_network(group$networks$treatment_1)
```

run_scNPM

Run Single-Cell Network Propensity Matching (scNPM)

Description

Estimate condition-specific gene networks using staged stochastic inference. The default first adapts networks and matching, aligns component labels to fixed initial mean anchors, and freezes a complete mapping after sustained stability. It then disables counterpart feedback, completes post-switch burn-in and retains final baseline-model samples. Failure to stabilize by the adaptation limit stops the run without producing final discoveries.

Usage

```

run_scNPM(
  goal_gene_mt,
  cell_t,
  K,
  ds_seed = 20250102,
  num_iter = 1000,
  num_save = 500,
  protein_input = NULL,
  ssp_v0 = 0.02,
  ssp_v1 = 1,
  ssp_l = 1,
  edge_prob_threshold = 0.6,
  edge_prob_adj = 0.75,
  epsilon_theta = 0.2,
  num_step_theta = 20,
  epsilon_lam = 0.01,
  num_step_lam = 10,
  ppi_prior_low = 0.05,
  ppi_prior_high = 0.5,
  network_covariates = c("all_reference", "legacy"),
  propensity_ridge = 1e-06,
  inference_mode = c("staged", "coupled"),
  adapt_min = NULL,
  adapt_max = NULL,
  freeze_patience = NULL,
  final_burnin = NULL,
  diagnostic_every = 10L,
  differential_method = c("auto", "bayes_fdr", "frequency"),
  fdr_level = 0.05,
  fdr_scope = c("frozen", "all_matchings"),
  fdr_min_draws = 200L,
  matching_method = c("lexicographic", "greedy"),
  fdr_rule = c("delta_threshold", "posterior_error"),
  group_fdr_level = 0.05,
  verbose = FALSE
)

```

Arguments

<code>goal_gene_mt</code>	Finite nonnegative numeric matrix or data frame, with genes in rows and cells in columns. At least two genes are required. Row names must be unique gene identifiers; missing names are generated.
<code>cell_t</code>	Finite numeric matrix or data frame, with cells in the same order as columns of <code>'goal_gene_mt'</code> . The first column is 0 (control) or 1 (treatment); both groups must be present. Remaining columns are numeric covariates, standardized across both groups. Encode categorical covariates numerically before calling this function.

K	Positive integer number of clusters in each condition, no larger than the number of distinct expression profiles in either condition.
ds_seed	Integer seed between 0 and 2147483646. Reproducibility requires the same inputs, software environment and OpenMP thread count.
num_iter	Positive integer total number of sampler iterations.
num_save	Positive integer number of final iterations retained, no greater than 'num_iter'.
protein_input	Optional PPI data frame, local TSV path, or TSV text. Required columns are 'node1', 'node2' and 'combined_score'. A leading '#' on the header is accepted. Positive scores indicate an interaction; duplicate pairs are combined, self-interactions and genes outside the expression matrix are ignored. 'NULL' assigns the lower prior to all edges.
ssp_v0	Positive spike standard deviation, smaller than 'ssp_v1'.
ssp_v1	Positive slab standard deviation.
ssp_l	Positive diagonal shrinkage parameter; the exponential-prior rate for precision diagonals is 'ssp_l / 2'.
edge_prob_threshold	Confidence threshold in (0.5, 1) for deciding whether an edge receives matching feedback; this is separate from the differential-network cutoff.
edge_prob_adj	Feedback multiplier in (0, 1]. A value of 1 leaves baseline edge probabilities unchanged by matching feedback.
epsilon_theta	Positive fixed HMC step size for latent expression.
num_step_theta	Positive integer number of HMC steps for latent expression.
epsilon_lam	Positive fixed HMC step size for dropout parameters.
num_step_lam	Positive integer number of HMC steps for dropout parameters.
ppi_prior_low	Baseline inclusion probability for edges without a PPI interaction (default 0.05).
ppi_prior_high	Baseline inclusion probability for PPI-supported edges (default 0.5). Requires '0 < ppi_prior_low <= ppi_prior_high <= 1 - 1e-6'.
network_covariates	Network features in the propensity model. "all_reference" (default) evaluates every cell against all current treatment networks followed by all control networks, using $(\theta_i - \mu_r)^T \Omega_r (\theta_i - \mu_r)$. "legacy" uses the single uncentered quadratic form from version 1.0.2. Other cell covariates are retained; matching_method selects the assignment rule.
propensity_ridge	Positive diagonal ridge stabilization for the pooled logistic IRLS fit (default 1e-6, also applied to the intercept). This is a numerical setting, not an empirically tuned penalty.
inference_mode	'staged' (default) or 'coupled' (continuous feedback; use greedy matching for version 1.0.3 behavior).
adapt_min	Earliest freezing iteration. NULL uses min(100, floor(adapt_max/2)), at least 2.
adapt_max	Latest adaptation iteration. NULL uses num_iter-num_save-final_burnin.

freeze_patience	Consecutive identical complete maps needed to freeze, at least 2. NULL uses $\min(20, \text{floor}(\text{adapt_max}/2))$, at least 2.
final_burnin	Minimum iterations after disabling feedback and before retention. NULL uses $\text{floor}((\text{num_iter}-\text{num_save})/3)$, at least 1.
diagnostic_every	Positive interval for final-stage J_{aug} diagnostics; adaptation diagnostics and matching histories are recorded every iteration.
differential_method	'auto' chooses 'bayes_fdr' for staged fits and descriptive 'frequency' selection for coupled fits. Explicit 'frequency' is available for staged comparisons; 'bayes_fdr' requires staged sampling.
fdr_level	Posterior directional error target in (0,1), default 0.05.
fdr_scope	'frozen' (default) uses the frozen correspondence. 'all_matchings' uses a conservative bound over all control counterparts.
fdr_min_draws	Minimum retained draws for FDR discoveries, at least 20, default 200. Fewer draws return probabilities and no selected discoveries.
matching_method	'lexicographic' (default) minimizes maximum pair SMD, then total SMD, with deterministic ties and no SMD cap. 'greedy' reproduces the previous assignment rule. A frozen map is not reoptimized.
fdr_rule	'delta_threshold' (default) scans all distinct absolute contrast cutoffs and retains the largest complete threshold set whose mean posterior directional error is at most fdr_level. 'posterior_error' retains the version 1.0.4 error-sorted prefix rule.
group_fdr_level	Group-network BFDR target in (0,1), default 0.05. Each condition/subtype is selected separately over its full gene panel.
verbose	Logical; FALSE (default) suppresses informational output. TRUE emits initialization, progress and timing through message conditions, including native sampler progress; suppressMessages() can suppress these. Warnings and errors are unaffected.

Details

Network features are centered distances to all $2 \times K$ current references, with pooled scaling; the legacy uncentered feature remains available. Default differential selection bounds the estimated posterior directional error over every possible control counterpart. This is model-based Bayesian error selection, not a finite-sample frequentist FDR guarantee. The coupled mode reproduces version 1.0.3 and uses descriptive frequency contrasts. The J_{aug} trace uses an augmented-data likelihood, a state-wise network prior kernel and total SMD. It is a diagnostic, not an optimized objective or the observed-data marginal likelihood; stochastic decreases are allowed.

Value

A named list containing all key analysis results (can be saved with 'save()').

runtime	MCMC runtime in minutes (difftime object)
K	Number of cell clusters used in analysis
G	Number of genes (rows in goal_gene_mt)
N	Number of cells (columns in goal_gene_mt)
lam0_est	Retained-draw means of lambda0 parameter (length = G)
lam1_est	Retained-draw means of lambda1 parameter (length = G)
mu_treat_est	Retained-draw summaries for treatment group (G x K matrix)
mu_ctrl_est	Retained-draw summaries for control group (G x K matrix)
invcov_treat_est	Retained-draw precision summaries for treatment group (G x G x K array)
invcov_ctrl_est	Retained-draw precision summaries for control group (G x G x K array)
edge_treat_est	Retained-draw edge frequencies for treatment group (G x G x K array)
edge_ctrl_est	Retained-draw edge frequencies for control group (G x G x K array)
group_treat_est	Cluster labels for treatment cells (length = number of treatment cells)
group_ctrl_est	Cluster labels for control cells (length = number of control cells)
pi_treat_est	Retained-draw mixing-weight means for treatment group (length = K)
pi_ctrl_est	Retained-draw mixing-weight means for control group (length = K)
theta_treat_est	Retained-draw latent-expression summaries for treatment group
theta_ctrl_est	Retained-draw latent-expression summaries for control group
theta_est	Retained-draw latent-expression summaries (cells x genes)
matching_result_df	Cluster matching table (treatment_cluster vs matched_control_cluster)
matching_method,	smd_cost_matrix, frozen_smd_cost_matrix,
matching_objective	Assignment rule, all K by K final-state pair costs, and selected maximum/total SMD. Frozen maps are retained even if final costs change.
smd_results	Standardized Mean Difference (SMD) values for each treatment cluster (computed on the propensity score)
smd_covariate_results	Per-covariate SMD matrix for matched cluster pairs (rows = treatment clusters, columns = the QC covariates in cell_t[, -1] plus the selected network features; NA for unmatched pairs). A value below 0.1 is a conventional balance diagnostic
match_map	Named vector mapping treatment clusters to matched control clusters
qf_global	Legacy uncentered per-cell quadratic form retained for compatibility. It enters the propensity model only in legacy mode.
network_covariates	Raw network-feature matrix from the final matching call. Columns are qf_ref_T1 through qf_ref_TK then qf_ref_C1 through qf_ref_CK in all-reference mode, or qf_network in legacy mode.

network_covariate_mode	Selected network feature mode.
propensity_scaling	Pooled centers and scales for the full predictor matrix, with other cell covariates before network features.
propensity_scores	Per-cell propensity scores from the final matching call
final_group_label	Final MCMC cluster label per cell (0-based: treatment cells 0..K-1, control cells K..2K-1; used by compute_qf_smd())
edge_diff_networks	Per-cluster networks selected by differential_method; solid/dashed indicates treatment/control enrichment. The matrix uses Rao-Blackwell contrasts for bayes_fdr and empirical frequency contrasts for frequency.
inference_mode, differential_method	Resolved inference and selection modes.
sampling_schedule	Freeze time, actual post-switch burn-in and retained interval.
stage_history, matching_history	Per-iteration phase, feedback status and aligned mapping history.
criterion_trace	Recorded augmented likelihood, network prior kernel, SMD loss and J_aug.
joint_edge_probabilities	Final all-pair 00/10/01/11 Rao-Blackwell probabilities, empirical 10/01 frequencies and batch means; slice (k-1)*K+c.
group_fdr	Per-condition/subtype BFDR selection from marginal Rao-Blackwell probabilities; NULL in coupled mode. Contains networks, masks, probabilities, summary and edge tables. Pass a named network to plot_group_network().
differential_fdr	Full directional error table, selected rows, estimated bound, scope and status; NULL for frequency mode.
Result	Retained sampler output, including parameter draws and selected trace summaries
protein_adjacency	Symmetric binary PPI adjacency matrix; diagonal entries are zero.
edge_prior	Per-edge baseline inclusion probabilities used at every edge update. Diagonal entries are unused.
provenance	Package version, input fingerprint, native-library fingerprint, execution environment, times and analysis settings.

Examples

```
# Small demonstration of staged inference; use longer runs for an analysis.
set.seed(7)
x <- matrix(rpois(6 * 40, lambda = 4), nrow = 6,
            dimnames = list(paste0("G", 1:6), NULL))
```

```

cells <- cbind(condition = rep(0:1, each = 20))
ppi <- data.frame(node1 = "G1", node2 = "G2", combined_score = 0.9)
fit <- run_scNPM(x, cells, K = 1, num_iter = 60, num_save = 20,
               adapt_min = 5, adapt_max = 15, freeze_patience = 3,
               final_burnin = 20, fdr_min_draws = 20,
               num_step_theta = 2, num_step_lam = 2,
               epsilon_theta = 0.01, epsilon_lam = 0.01,
               protein_input = ppi)
fit$matching_result_df
fit$group_fdr$summary
compute_qf_smd(fit)

```

select_differential_edges

Select Differential Edges by Posterior Directional Error

Description

Uses the final baseline-model Rao-Blackwell probabilities from a staged run. The selection unit is one treatment cluster and unordered gene pair; all K clusters form a single family. The most supported direction under the frozen map is reported. For ‘all_matchings’, its error is bounded by the largest posterior error over all K possible control counterparts. This is a conservative bound for every fixed bijection, not a posterior distribution over matchings and not a frequentist FDR guarantee.

Usage

```

select_differential_edges(
  result,
  fdr_level = 0.05,
  scope = c("frozen", "all_matchings"),
  min_draws = 200L,
  rule = c("delta_threshold", "posterior_error")
)

```

Arguments

result	Output of a staged [run_scNPM()] fit.
fdr_level	Target posterior expected directional FDP in (0, 1).
scope	‘frozen’ (default) uses the selected matching. ‘all_matchings’ bounds errors over every control counterpart for sensitivity analysis.
min_draws	Minimum retained iterations before producing discoveries, at least 20, default 200. This guard does not establish convergence.
rule	‘delta_threshold’ scans strict absolute contrast thresholds without splitting tied scores; ‘posterior_error’ uses the version 1.0.4 error prefix.

Details

The selected set has mean estimated directional error at most ‘fdr_level’. Monte Carlo approximation, label identification and model assumptions still apply. Edge states refer to continuous spike/slab allocations, not exact zero precision entries. Batch MCSE is a diagnostic, not a confidence bound or convergence certificate.

Value

A list with threshold (strict $|\Delta| > \text{threshold}$), threshold_curve, rule, and the full edge table, selected rows, estimated error bound, target, scope, sample count and status. Each table row has an ‘error_bound’ for that edge and a ‘frozen_error’ for its reported pairing. ‘estimated_error_bound’ is the mean bound across selected rows.

select_group_edges	<i>Select Group Networks by Posterior Expected False Discovery Proportion</i>
--------------------	---

Description

Reconstructs marginal Rao-Blackwell slab probabilities from final-stage joint probabilities: $p_{10} + p_{11}$ for treatment and $p_{01} + p_{11}$ for control. Each condition and subtype is a separate family of $G*(G-1)/2$ unordered pairs. Probabilities are sorted decreasingly, complete tie blocks are evaluated, and the largest prefix with mean error $1-p$ at most fdr_level is retained. Selection uses $p \geq \text{cutoff}$; an empty set has cutoff Inf and BFDR zero.

Usage

```
select_group_edges(result, fdr_level = 0.05, min_draws = 200L)
```

Arguments

result	Output of a staged [run_scNPM()] fit with joint probabilities.
fdr_level	BFDR target in (0,1), default 0.05, separately for each network.
min_draws	Minimum retained iterations, at least 20, default 200. Below this value, probabilities and curves are returned with empty masks.

Details

Selection is performed on the full gene panel before any subnetwork is extracted. Returned masks should be shared by plotting and sensitivity calculations. BFDR applies to each original network, not to a subsequent union over subtypes. Edge states are continuous spike/slab allocations. This estimates posterior expected FDP under the baseline model; it is not a uniform frequentist FDR guarantee. Existing staged 1.0.5 fits with saved joint probabilities can be processed without rerunning the sampler.

Value

A list with 'networks', 'summary', 'table', 'selected', 'probabilities', 'masks', 'target', 'status' and draw counts. Each named network (for example 'treatment_1') contains 'probability_matrix', 'mask', 'cutoff', 'estimated_bfdr', 'threshold_curve', 'table' and 'selected'. 'probabilities' and 'masks' each contain treatment/control $G \times G \times K$ arrays. 'probability_matrix' carries a saved selection attribute for plotting.

Examples

```
# Synthetic retained posterior summaries for one matched pair.
# These values illustrate selection; no MCMC is run in this example.
gn <- paste0("G", 1:3)
p <- matrix(c(0,.99,.97,.99,0,.60,.97,.60,0), 3,
            dimnames = list(gn, gn))
z <- array(0, c(3,3,1))
p10 <- array(p, c(3,3,1)); p00 <- array(1-p, c(3,3,1))
diag(p00[, , 1]) <- 0
example_fit <- list(
  K = 1L, G = 3L, gene_names = gn, inference_mode = "staged",
  matching_result_df = data.frame(treatment_cluster=1L, matched_control_cluster=1L),
  sampling_schedule = list(feedback_disabled_for_retained=TRUE, retained_draws=500L),
  joint_edge_probabilities = list(p00=p00, p10=p10, p01=z, p11=z)
)
group <- select_group_edges(example_fit)
group$summary
```

Index

cell, [2](#)
cols_to_MR, [3](#)
cols_to_PR, [3](#)
compute_qf_smd, [4](#)

dgTMatrix, [6](#)

goal_gene, [5](#)
goal_gene_mt, [5](#)

mt, [6](#)

plot_diff_network, [7](#)
plot_group_network, [8](#)

run_scNPM, [2](#), [9](#)

select_differential_edges, [15](#)
select_group_edges, [16](#)