Package: NetSurvProx (via r-universe)

June 9, 2026

Type Package

Title 'NetSurvProx': Network-Based Survival Analysis via Proximal Methods

Version 1.0.0

Maintainer Maura Mecchi <maura.mecchi@unibas.it>

Description Introduces a novel network-constrained survival analysis framework for variable selection and parameter estimation in penalized survival models with convex penalties. The package extends two classical survival models, the Cox Proportional Hazards (PH) model and the Accelerated Failure Time (AFT) model, by incorporating prior biological knowledge from curated interaction networks (e.g., KEGG) into a double-penalty framework. The first penalty enforces variable selection through a LASSO penalty, while the second preserves gene-gene correlations by incorporating Laplacian-based constraints, ensuring that biologically relevant network structures are maintained. Using censored survival data, the method enables the identification of predictive biomarkers and pathways with potential relevance for target therapies. Model estimation is performed via proximal optimization algorithms combined with cross-validation for reliable tuning. To enhance interpretability, dedicated utility functions are implemented to consolidate results, yielding biologically coherent insights that can support personalized medicine and contribute to improved patient outcomes.

Depends R (>= 4.3)

Imports AnnotationDbi, curl, cvTools, dplyr, flexsurv, foreach, ggplot2, ggpubr, glmnet, grDevices, Hmisc, httr, igraph, magic, openxlsx, RColorBrewer, rmarkdown, survAUC, survival, survminer,

Suggests knitr, org.Hs.eg.db, plotly, scales, sessioninfo, stringr, visNetwork

VignetteBuilder knitr

License GPL (>= 3)

Encoding UTF-8
RoxygenNote 7.3.3
LazyData true
LazyDataCompression bzip2
NeedsCompilation no
Author Maura Mecchi [aut, cre], Antonella Iuliano [aut]
Config/pak/sysreqs zlib1g-dev
Repository https://cran.r-universe.dev
Date/Publication 2026-06-09 08:43:52 UTC
RemoteUrl https://github.com/cran/NetSurvProx
RemoteRef HEAD
RemoteSha 210154a09bc18f64e8f6fed382a097a00743e889

Contents

`CreateNetwork` *Laplacian Matrix for Prior Biological Knowledge in Network Constraint*

Description

Builds a Laplacian network penalty based on a prior weighted graph. It encourages coefficients corresponding to connected covariates to behave similarly: if two covariates are strongly connected in the network, their estimated coefficients tend to be either both close to zero or both nonzero. In this way, the penalty promotes smoothness and structural coherence across related variables.

Usage

```
CreateNetwork(
  X,
  Y = NULL,
  delta = NULL,
  doid = NULL,
  tissue = NULL,
  disease_file = NULL,
  tissue_file = NULL,
  cache = FALSE,
  cache_dir = NULL,
  choice = 1,
  model = NULL,
  dist = NULL,
  verbose = FALSE
)
```

Arguments

X	Numeric matrix of standardized covariates.
Y	Numeric vector of observed survival times (log-transformed under AFTNet), required for choice = 2.
delta	Integer vector of censoring indicators (1 = event, 0 = censored), required for choice = 2.
doid	Character string specifying Disease Ontology ID ("DOID:XXXX"), used only if disease_file is not provided.
tissue	Character string specifying tissue name, used to retrieve the tissue-specific network from HumanBase, used only if tissue_file is not provided.
disease_file	Character string specifying optional path to a tab-delimited file containing disease-associated genes (columns: entrez_id, standard_name, and score).
tissue_file	Character string specifying optional path to a tab-delimited file with tissue-specific gene interactions (columns: gene1, gene2, and score).
cache	Logical value; if TRUE, downloaded HumanBase files are cached for reuse in cache_dir. If FALSE (default), files are downloaded for the current session only.
cache_dir	Character string specifying a directory used to cache downloaded HumanBase files (when cache = TRUE).
choice	Value specifying the choice for the signs of the adjacency matrix • 1 (default): for correlation-based signs. • 2: for ridge-based signs.
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet") required only for choice = 2.
dist	Character string specifying the AFTNet distribution. Must be one of "weibull", "lognormal", or "loglogistic" (required only for choice = 2).
verbose	Logical value, if TRUE progress messages are printed.

Details

This prior network is represented by a weighted graph where each vertex corresponds to a covariate and the edges describe relationships between covariates. The edge weights are stored in an adjacency matrix A, which has zeros on its diagonal. The degree matrix D contains on its diagonal the sum of the absolute edge weights connected to each vertex. The Laplacian matrix is defined as $L = D - W$, where W is the weighted matrix estimated from A. Two strategies can be used.

- Correlation-based signs (`choice = 1`): the sign of an edge is set according to the Pearson correlation between the two corresponding covariates.
- Ridge-based signs (`choice = 2`): the sign of an edge is determined by the signs of ridge regression coefficients obtained from a penalized survival model. This ridge estimator provides stable coefficient estimates in high-dimensional settings. For the Cox model the ridge fit is obtained via `glmnet::glmnet()`), while for the AFT model via `survival::survreg()`).

The framework is used to construct a disease-specific gene interaction network, where edges represent biological relationships between genes relevant to a given cancer and tissue type.

Internally, the function relies on helper routines (see `RepositoryDisease` and `RepositoryTissue`) to retrieve biological prior information from the HumanBase database. These datasets are combined to construct a disease- and tissue-specific adjacency matrix that defines the structure of the Laplacian penalty. User-provided files with the same format can be supplied to bypass the download step.

Value

A list with two elements:

- `disease_genes`: data frame of disease genes used in the network.
- `L`: final Laplacian matrix.

Note

If tissue-specific or disease-specific files are not provided, the function downloads the relevant data from HumanBase. In this case, an active internet connection is required. Moreover, not all DOIDs and tissues are present in the HumanBase repository. f the requested is not available, the function may return an empty list.

Examples

```
data(LUADdataset)

net <- CreateNetwork(
          LUADdataset$X_train,
          doid    = "DOID:1324",
          tissue  = "lung",
          choice  = 1,
          verbose = TRUE)

L   <- net$L                          # final laplacian matrix
```

```
disease_genes <- net$disease_genes    # disease genes and scores
```

CvNet *Cross-validated Linear Predictors Approach for* COXNet *and* AFTNet

---

Description

Performs K-fold cross-validation to select the optimal regularization parameter λ for penalized survival models (COXNet, AFTNet) estimated via ProxGDNet. The criterion is based on cross-validated linear predictors and negative (partial) log-likelihood.

Usage

```
CvNet(
  X,
  Y,
  delta,
  L = NULL,
  lambda,
  alpha,
  model = NULL,
  dist = NULL,
  sigma = NULL,
  nfolds = 5,
  seed = 2026,
  value = 2,
  niter = 1000,
  conv = 0.001,
  parallel = TRUE,
  ncore_max = 5,
  verbose = FALSE
)
```

Arguments

X	Numeric matrix of standardized covariates.
Y	Numeric vector of observed survival times (log-transformed under AFTNet).
delta	Integer vector of censoring indicators (1 = event, 0 = censored).
L	Optional positive semi-definite, symmetric, and diagonally dominant Laplacian matrix encoding prior network information (see CreateNetwork for details). If NULL, no network-based penalization is applied.
lambda	Numeric vector of candidate tuning parameters (in descending order).

alpha	Numeric parameter controlling the convex combination of the two penalty terms (value in [0,1]).
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet").
dist	Character string specifying the error distribution of AFTNet model. Must be one of "weibull", "lognormal", or "loglogistic".
sigma	Positive numeric scalar representing the scale parameter of the error distribution in AFTNet model.
nfolds	Number of cross-validation folds (default: 5).
seed	Random seed for reproducibility (default: 2026).
value	Numeric scalar greater than 1 specifying the multiplicative factor used to increase the step-size constant during backtracking line search (default: 2).
niter	Maximum number of proximal gradient iterations (default: 1000).
conv	Convergence tolerance for proximal gradient (default: 1e-3).
parallel	Logical value, whether to use parallel processing (default: TRUE).
ncore_max	Maximum number of cores for parallel processing over cross validation (default: 5).
verbose	Logical value, if TRUE progress messages are printed (default: FALSE).

Details

The dataset is split into K folds. For each fold, the model is trained on K-1 folds, and evaluated on the held-out fold. The cross-validated linear predictor is computed as

$$\hat{\eta}_i^{CV} = \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}_\lambda^{(-k)}$$

for COXNet, or the cross-validated standardized residual as

$$\hat{e}_i^{CV} = \frac{y_i - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}_\lambda^{(-k)}}{\hat{\sigma}}$$

for AFTNet, and used to evaluate the cross-validation criterion over a grid of λ values.

The optimal parameter is selected according to:

- the minimum CV error (lambda.min).
- the largest λ within one standard error of the minimum (lambda.1se).

Value

An object of class "cv.out" containing:

- cv.err.linPred: CV error for each value of λ.
- cv.err.obj: estimated standard error associated with each value of CV error per fold.
- lambda.grid: grid of regularization parameters values.
- lambda.min: value of λ minimizing the CV error.
- ind.lambda.min: indices of lambda.min.

- `lambda.1se`: largest λ within one standard error of the minimum.
- `ind.lambda.1se`: indices of `lambda.1se`.
- `cvup`: upper error curve.
- `cvlo`: lower error curve.

Note

Computation can be performed sequentially (`parallel: FALSE`), or in parallel (`parallel: TRUE`) using `parLapply`. The number of cores is automatically determined based on system availability, number of folds and user-specified maximum `ncore_max`.

See Also

- `PlotCvNet` for visualization of the obtained cross-validation curve.
- `ProxGDNet` for proximal network-penalized gradient descent algorithm details.

`Enrichment` *Pathway Enrichment (Over-representation Analysis)*

Description

Performs pathway enrichment analysis to evaluate whether a set of genes is over-represented in one or more pathways compared to a background set of genes. For each pathway, it calculates the number of observed genes, the Fisher's exact test p-value, and FDR-adjusted p-values. Significant pathways (`padj < 0.05`) are marked with `Yes` in the `highlight` column.

Usage

```
Enrichment(
  genes,
  pathway_df,
  background_genes = NULL,
  min_genes = 2,
  top_n = 10,
  out_file = NULL
)
```

Arguments

`genes`	Character vector specifying the list of selected gene symbols.
`pathway_df`	Data frame with at least the following columns: • `pathway`: pathway identifier. • `gene`: gene symbol belonging to the pathway. • `name`: optional descriptive name for the pathway.

`background_genes`
: Character vector specifying background gene set. If `NULL` (default), the union of `genes` and all genes in `pathway_df` is used.

`min_genes`
: Numeric value specifying the minimum number of background genes that a pathway must have to be considered (`default: 2`).

`top_n`
: Numeric value specifying the number of top pathways sorted by adjusted p-value to return (`default: 10`).

`out_file`
: Character string specifying the path to save the enrichment results as an Excel file (.xlsx). If `NULL` (default), the results are not written to disk.

Details

The function implements an over-representation analysis (ORA) workflow:

1. Intersects the input gene list with a background set (user-provided or derived from all pathway genes).
2. Filters pathways to retain only those with at least `min_genes` present in the background.
3. Performs Fisher's exact test for each pathway to assess over-representation.
4. Adjusts p-values using the false discovery rate (FDR) method.
5. Identifies significantly enriched pathways (`padj < 0.05`) and marks them in the `highlight` column.
6. Selects the top `top_n` pathways for visualization in dashboards or plots.

The results are automatically saved as an Excel file `Enrichment_results.xlsx` and are used by `PathwayDashboard` to display enrichment results interactively in the dedicated panel.

Value

A list containing:

- `results`: Full enrichment table with p-values and FDR correction, including `pathway`, `nGenes` (number of genes for pathway), `pval`, `padj`, `highlight` (`Yes/No` if the pathway is enriched), `name`.
- `bar_data`: Top `top_n` enriched pathways.

See Also

`PathwayDashboard` for interactive visualization of enrichment results.

LUADdataset *Example Dataset for Network-Based Survival Analysis*

Description

A pre-processed dataset containing clinical survival information and gene expression covariates for Lung Adenocarcinoma (TCGA-LUAD). This dataset allows users to bypass the computationally intensive download and preprocessing pipeline, providing immediate access to the covariate matrix, survival outcomes, and censoring indicators.

Usage

```
data(LUADdataset)
```

Format

A list with the following components.

- `X_train` : numeric matrix of training covariates.
- `X_test` : numeric matrix of testing covariates.
- `Y_train` : numeric vector of observed training survival times.
- `Y_test` : numeric vector of observed testing survival times.
- `delta_train` : integer vector of training censoring indicators.
- `delta_test` : integer vector of testing censoring indicators.

Details

Gene expression data (RNA-seq) were obtained from the LinkedOmics portal and processed to construct:

- screened gene expression matrix `X` (samples × genes),
- observed survival times `Y` (real scale),
- censoring indicators `delta` (1 = event, 0 = censored).

The screening was performed using the BMD method (see `VariableScreening`) focusing on disease-associated genes retrieved for `doid = "DOID:1324"` via `RepositoryDisease`.

The dataset is pre-partitioned into an 70% training set for model estimation and a 30% testing set for validation.

Source

https://linkedomics.org/data_download/TCGA-LUAD/

Metrics *Performance Metrics for Survival Models*

Description

Computes a variety of performance metrics for survival model supporting both real-data evaluation and simulation studies.

Usage

```
Metrics(
  Y_train = NULL,
  delta_train = NULL,
  X_test = NULL,
  Y_test = NULL,
  delta_test = NULL,
  beta_est,
  beta_true = NULL,
  model = NULL,
  p_active = NULL,
  times_auc = NULL,
  metrics = NULL
)
```

Arguments

`Y_train`	Numeric vector of observed training survival times (log-transformed under `"AFTNet"`).
`delta_train`	Integer vector of training censoring indicators (1 = event, 0 = censored).
`X_test`	Numeric matrix of testing covariates standardized using the training data.
`Y_test`	Numeric vector of observed testing survival times (log-transformed under `"AFTNet"`).
`delta_test`	Integer vector of testing censoring indicators (1 = event, 0 = censored).
`beta_est`	Numeric vector of estimated regression coefficients obtained from the training set.
`beta_true`	Optional numeric vector of true regression coefficients. Required only for simulation-based metrics (FPR, FNR, PMSE).
`model`	Character string specifying the fitted survival model (`"COXNet"`, or `"AFTNet"`).
`p_active`	Integer scalar specifying the number of truly active covariates, required only when `metrics` includes `"FPR"` or `"FNR"` and `beta_true` is supplied.
`times_auc`	Optional numeric vector of time points at which the time-dependent AUC is evaluated. If `NULL` (default), empirical quantiles of `Y_test` are used.
`metrics`	Character vector specifying the performance measures to compute. Allowed values: • `"PredRisk"` - Predicted Risk or expected survival time,

- "CIndex" - Harrell's concordance index,
- "FPR" - False Positive Rate,
- "FNR" - False Negative Rate,
- "NSR" - Number of Selected variables Rate,
- "PMSE" - Predictive Mean Square Error,
- "AUC" - time-dependent AUC.

Details

The predicted quantity depends on the model type:

- For COXNet, PredRisk is the hazard ratio.
- For AFTNet, PredRisk is proportional to the expected survival time.

Harrell's concordance index is computed using rcorr.cens. The time-dependent AUC is computed using Uno's estimator via AUC.uno at the specified time points.

The metrics FPR, FNR, and PMSE are defined only in simulation settings because they require knowledge of the true regression coefficients. When beta_true is not provided, these metrics are returned as NA if requested. All other metrics can be computed for both simulated and real datasets.

Value

A named list containing the requested performance metrics.

Note

Scalar metrics are returned as numeric values, PredRisk as a numeric vector of predicted risk scores, and time-dependent AUC values as separate list elements with names of the form "AUC_t_<time>".

NetSurvProx *NetSurvProx Complete Routine*

Description

Fits network-constrained penalized survival models (COXNet and AFTNet) to identify prognostic signature genes and build a Prognostic Index (PI). The model is trained on a training dataset by incorporating both Laplacian constraints and LASSO regularization, with optional feature standardization. The tuning parameters are jointly selected through cross-validation. An optimal cutoff for the PI is estimated from the training data to enable prognostic stratification. Predictive performance is subsequently evaluated on an independent testing dataset. Model assessment includes survival curve analyses and visualization. Predictive accuracy is quantified using selected metrics.

Usage

```
NetSurvProx(
  X_train,
  Y_train,
  delta_train,
  X_test,
  Y_test,
  delta_test,
  L = NULL,
  standardize_train = TRUE,
  standardize_test = TRUE,
  model = NULL,
  dist = NULL,
  select_lambda = TRUE,
  alpha_grid = c(0.3, 0.5, 0.7),
  nlambda = 50,
  lambda_ratio = 0.01,
  nfolds = 5,
  method = NULL,
  probs = seq(0.25, 0.8, by = 0.05),
  cutoffplot = FALSE,
  seed = 2026,
  value = 2,
  niter = 1000,
  conv = 0.001,
  parallel_cv = TRUE,
  plotCV = FALSE,
  colors_pcv = NULL,
  errorbar = FALSE,
  ncore_max = 5,
  p_active = NULL,
  times_auc = NULL,
  beta_true = NULL,
  metrics = NULL,
  verbose = FALSE,
  palette = NULL,
  plot_test = FALSE
)
```

Arguments

X_train	Numeric matrix of training covariates standardized (possibly screened using screen_vars, see VariableScreening).
Y_train	Numeric vector of observed training survival times (log-transformed under AFTNet).
delta_train	Integer vector of training censoring indicators (1 = event, 0 = censored).
X_test	Numeric matrix of testing covariates.
Y_test	Numeric vector of observed testing survival times (log-transformed under AFTNet).

`delta_test` Integer vector of testing censoring indicators (1 = event, 0 = censored).

`L` Optional positive semi-definite, symmetric, and diagonally dominant Laplacian matrix encoding prior network information (see `CreateNetwork`). If `NULL`, no network-based penalization is applied.

`standardize_train` Logical value indicating whether to standardize the training matrix: if `TRUE` (default), each column is centered to have mean 0 and scaled to have unit variance, if `FALSE`, the matrix is assumed pre-standardized by the user.

`standardize_test` Logical value indicating whether to standardize `X_test` with respect to `X_train` (`default: TRUE`).

`model` Character string specifying the fitted survival model (`"COXNet"`, or `"AFTNet"`).

`dist` Character string specifying the `AFTNet` distribution. Must be one of `"weibull"`, `"lognormal"`, or `"loglogistic"`.

`select_lambda` Logical value, if `TRUE` (default) uses `lambda.min`, otherwise `lambda.1se`.

`alpha_grid` Numeric vector specifying the candidate values for α in `[0,1]` (`default: c(0.3, 0.5, 0.7)`).

`nlambda` Numeric value specifying the number of candidate values for λ in the grid (`default: 50`).

`lambda_ratio` Numeric value giving the ratio of minimum to maximum λ in the grid (`default: 0.01`).

`nfolds` Numeric value of folds performed for tuning optimal parameters (`default: 5`).

`method` Character string specifying the cutoff selection method (`"median"` or `"minpvalue"`, see `OptimalPICutoff`).

`probs` Vector of probabilities used when `method = "minpvalue"` to generate candidate cutoffs based on quantiles of the PI (`default: probs = seq(0.25, 0.80, by = 0.05)`).

`cutoffplot` Logical value indicating whether survival curves should be produced (`default: FALSE`).

`seed` Random seed for reproducibility (`default: 2026`).

`value` Numeric scalar greater than 1 specifying the multiplicative factor used to increase the step-size constant during backtracking line search in `ProxGDNet` (`default: 2`).

`niter` Maximum number of iterations for `ProxGDNet` (`default: 1000`).

`conv` Convergence tolerance for ProxGDNet (`default: 1e-3`).

`parallel_cv` Logical value whether to use parallel processing for `CvNet` (`default: TRUE`).

`plotCV` Logical value indicating whether CV curves should be shown (`default: FALSE`).

`colors_pcv` Optional named list of colors for CV plot (see `CvNet`).

`errorbar` Logical value, if `TRUE` the CV plot includes vertical error bars representing 1se of the CV error (`default: FALSE`).

`ncore_max` Maximum number of cores for parallel processing over CV (`default: 5`).

p_active	Numeric value indicating the number of truly active covariates (required for FPR/FNR computation in simulation settings).
times_auc	Numeric vector of time points for time-dependent AUC. If NULL (default), quantiles of Y_test are used.
beta_true	Numeric vector of true coefficients (used only for simulated data).
metrics	Character vector specifying performance Metrics to compute. For real datasets: "CIndex", "NSR", "AUC". For simulated datasets (in addition): "FPR", "FNR", "PMSE".
verbose	Logical value, if TRUE progress messages are printed (default: FALSE).
palette	Optional character vector of length 2 specifying colors used for the survival curves. For "COXNet", colors correspond to high- and low-risk groups. For "AFTNet", colors correspond to short- and long-survival groups. If NULL, default colors are used.
plot_test	Logical value, if TRUE returns the combined survival plot with validation results (default: FALSE).

Value

An object of class NetSurvProx containing:

- fit_training: training results (see NetSurvProx_Training).
- fit_testing: testing results (see NetSurvProx_Testing).

Examples

```
# - Simulate 40 TFs, each regulating 10 targets with a independent structure -

targets <- 10

n <- 165

simul_data <- Simulations(
    n = n, r = 40, targets = targets, p_active = 40,
    rho = 0.70, rate = 0.50, b_true = c(0.8, 1.2, -1.2, -0.8),
    nsimul = 1, model = "AFTNet", baseline = "lognormal",
    sigma_true = 1, shared_scheme = NULL, choice = 1,
    save = FALSE, save_path = NULL, seed = 2026, verbose = TRUE)

X     <- simul_data$X_list[[1]]
Y     <- simul_data$time_list[[1]]   # generated in log-scale
delta <- simul_data$delta_list[[1]]
L     <- simul_data$L_list[[1]]

beta_true <- as.vector(unlist(simul_data$beta))

#  - Split the dataset (training/testing sets) -

set.seed(2026)
```

```
  train_idx <- sample(seq_len(n), size = floor(0.7 * n))

  X_train     <- X[train_idx,]
  Y_train     <- Y[train_idx]
  delta_train <- delta[train_idx]

  X_test     <- X[-train_idx,]
  Y_test     <- Y[-train_idx]
  delta_test <- delta[-train_idx]

# - Fitting LogNormal AFTNet -

  out <- NetSurvProx(
              X_train, Y_train, delta_train, X_test, Y_test, delta_test,
              L = L, standardize_train = TRUE, standardize_test = TRUE,
              model = "AFTNet", dist = "lognormal", select_lambda = TRUE,
              alpha_grid = 0.5, nlambda = 50, lambda_ratio = 0.1,
              nfolds = 5, method = "minpvalue", probs = seq(0.25, 0.80, by = 0.05),
              cutoffplot = FALSE, seed = 2026, value = 2, niter = 1000, conv = 1e-3,
             parallel_cv = FALSE, plotCV = FALSE, colors_pcv = NULL, errorbar = FALSE,
              ncore_max = 1, p_active = 40, times_auc = NULL, beta_true = beta_true,
              metrics = "CIndex", verbose = FALSE, palette = NULL, plot_test = FALSE)

# - Results -

  data.frame(out$fit_testing$performance)
```

NetSurvProx_Testing *NetSurvProx Testing Routine*

Description

Evaluates predictive performance of a fitted `COXNet` or `AFTNet` model on an independent testing set. The function computes the Prognostic Index (PI) using the selected signature genes and the optimal cutoff obtained from the training phase, generates survival curves, PI distribution plots, and calculates specified performance metrics.

Usage

```
NetSurvProx_Testing(
  X_train = NULL,
  standardize = TRUE,
  Y_train = NULL,
  delta_train = NULL,
  X_test,
  Y_test,
```

```
  delta_test,
  model = NULL,
  dist = NULL,
  beta,
  beta_true = NULL,
  opt_cutoff,
  p_active = NULL,
  times_auc = NULL,
  metrics = NULL,
  verbose = FALSE,
  plot = FALSE,
  palette = NULL
)
```

Arguments

X_train	Numeric matrix of training covariates (used only to scale X_test when standardize = TRUE).
standardize	Logical value indicating whether to standardize X_test with respect to X_train (default: TRUE).
Y_train	Numeric vector of observed training survival times (log-transformed under AFTNet). Required only for time-dependent AUC computation.
delta_train	Integer vector of training censoring indicators (1 = event, 0 = censored). Required only for time-dependent AUC computation.
X_test	Numeric matrix of testing covariates.
Y_test	Numeric vector of observed testing survival times (log-transformed under AFTNet).
delta_test	Integer vector of testing censoring indicators (1 = event, 0 = censored).
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet").
dist	Character string specifying the AFTNet distribution. Must be one of "weibull", "lognormal", or "loglogistic".
beta	Numeric vector of regression coefficients estimated on the training set.
beta_true	Numeric vector of true coefficients (used only for simulated data).
opt_cutoff	Numeric value used to split the PI into two prognostic groups.
p_active	Numeric value indicating the number of truly active covariates (required for FPR/FNR computation in simulation settings).
times_auc	Numeric vector of time points for time-dependent AUC. If NULL (default), quantiles of Y_test are used.
metrics	Character vector specifying performance metrics to compute. For real datasets: "CIndex", "NSR", "AUC". For simulated datasets (in addition): "FPR", "FNR", "PMSE".
verbose	Logical value, if TRUE progress messages are printed (default: FALSE).
plot	Logical value, if TRUE returns the combined survival plot (default: FALSE).
palette	Optional character vector of length 2 specifying colors used for the survival curves. For "COXNet", colors correspond to high- and low-risk groups. For "AFTNet", colors correspond to short- and long-survival groups. If NULL, default colors are used.

Details

The testing set must be independent from the training set used in NetSurvProx_Training. When standardize = TRUE, X_test is standardized using the mean and standard deviation of X_train. Only covariates with non-zero coefficients in beta are retained for PI computation.

Prognostic stratification is performed using ValidationPI, producing:

- Kaplan–Meier curves and log-rank test for COXNet.
- Parametric survival curves and likelihood ratio test for AFTNet.
- PI distribution plots by prognostic group.

Value

A list containing:

- df: data frame with PI (computed for each subject), Y, delta, and groupRisk (prognostic group assigned based on opt_cutoff).
- p_value: from the log-rank test (COXNet) or likelihood ratio test (AFTNet).
- performance: named list with the requested performance metrics.

See Also

- Metrics for available performance metrics options.
- NetSurvProx_Training for training routine.
- OptimalPICutoff for opt_cutoff estimation.
- ValidationPI for PI validation and optional plot.

NetSurvProx_Training *NetSurvProx Training Routine*

Description

Trains penalized regression methods (COXNet or AFTNet) to incorporate gene regulatory relationships and select signature genes using the training set. Regularization parameters are selected via cross-validation, and an optimal Prognostic Index (PI) cutoff is determined for risk stratification (COXNet) or for survival time stratification (AFTNet). The procedure includes optional feature standardization and simultaneous selection of the regularization parameters for the Laplacian constraint and the Lasso penalty.

Usage

```
NetSurvProx_Training(
  X_train,
  Y_train,
  delta_train,
  L = NULL,
  model = NULL,
  dist = NULL,
  select_lambda = TRUE,
  alpha_grid = c(0.3, 0.5, 0.7),
  nlambda = 50,
  lambda_ratio = 0.01,
  nfolds = 5,
  method = NULL,
  probs = seq(0.25, 0.8, by = 0.05),
  cutoffplot = FALSE,
  seed = 2026,
  value = 2,
  niter = 1000,
  conv = 0.001,
  parallel = TRUE,
  plotCV = FALSE,
  colors_pcv = NULL,
  errorbar = FALSE,
  ncore_max = 5,
  standardize = TRUE,
  verbose = FALSE,
  palette = NULL
)
```

Arguments

X_train	Numeric matrix of training covariates standardized (possibly screened using screen_vars).
Y_train	Numeric vector of observed training survival times (log-transformed under AFTNet).
delta_train	Integer vector of training censoring indicators (1 = event, 0 = censored).
L	Optional positive semi-definite, symmetric, and diagonally dominant Laplacian matrix encoding prior network information. If NULL, no network-based penalization is applied.
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet").
dist	Character string specifying the AFTNet distribution. Must be one of "weibull", "lognormal", or "loglogistic".
select_lambda	Logical value, if TRUE (default) uses lambda.min, otherwise lambda.1se.
alpha_grid	Numeric vector specifying the candidate values for α in [0,1] (default: c(0.3, 0.5, 0.7)).

nlambda	Numeric value specifying the number of candidate values for λ in the grid (default: 50).
lambda_ratio	Numeric value giving the ratio of minimum to maximum λ in the grid (default: 0.01).
nfolds	Number of cross-validation folds (default: 5).
method	Character string specifying the cutoff selection method ("median" or "minpvalue").
probs	Vector of probabilities used when method = "minpvalue" to generate candidate cutoffs based on quantiles of the PI (default: probs = seq(0.25, 0.80, by = 0.05)).
cutoffplot	Logical value indicating whether survival curves should be produced (default: FALSE).
seed	Random seed for reproducibility (default: 2026).
value	Numeric scalar greater than 1 specifying the multiplicative factor used to increase the step-size constant during backtracking line search (default: 2).
niter	Maximum number of iterations for ProxGDNet (default: 1000).
conv	Convergence tolerance for ProxGDNet (default: 1e-3).
parallel	Logical value whether to use parallel processing for CvNet (default: TRUE).
plotCV	Logical value indicating whether cross-validation curves should be shown (default: FALSE).
colors_pcv	Optional named list of colors: • line: colorof the cross-validation error curve. • points: color of observed CV error evaluations. • min: color of the vertical line indicating lambda.min. • one_se: color of the vertical line indicating lambda.1se. If NULL, a default color palette is used.
errorbar	Logical value, if TRUE the CV plot includes vertical error bars representing 1SE of the CV error (default: FALSE).
ncore_max	Maximum number of cores for parallel processing over CV (default: 5).
standardize	Logical value indicating whether to standardize the input matrix: if TRUE (default), each column is centered to have mean 0 and scaled to have unit variance, if FALSE, the matrix is assumed pre-standardized by the user.
verbose	Logical value, if TRUE progress messages are printed (default: FALSE).
palette	Optional character vector of length 2 specifying colors used for the survival curves. For "COXNet", colors correspond to high- and low-risk groups. For "AFTNet", colors correspond to short- and long-survival groups. If NULL, default colors are used.

Details

The function performs joint tuning for regularization parameters: a grid of α values in (0, 1) is constructed, and for each candidate computes corresponding λ grids via cross-validation using the negative (partial for COXNet) log-likelihood's gradient.

Parallel computation is supported to improve efficiency.

Value

A list containing:

- `alpha.opt`: numeric value of optimal alpha.
- `lambda.opt`: numeric value of optimal lambda.
- `beta`: estimated regression coefficients.
- `index.nonzerobeta`: index of non-zero beta.
- `lambda.min`: value of λ minimizing the CV error.
- `lambda.1se`: largest λ within one standard error of the minimum.
- `cutoff.opt`: numeric value of optimal prognostic index cutoff.
- `lambda.grid`: grid of regularization parameters values.
- `cv.err.linPred`: cross-validated error for each value of λ.
- `cv.err.obj`: estimated standard error associated with each value of CV error.
- `full_summary`: data.frame as summary of CV results for all tested `alpha` values.

See Also

- `CreateNetwork`: for `L` matrix computation.
- `CvNet`: for CV and parallel processing details.
- `PlotCvNet`: for cross-validation plot.
- `OptimalPICutoff`: for the optimal cutoff value to stratify observations.
- `ProxGDNet`: for proximal network-penalized gradient descent algorithm details.
- `VariableScreening`: for the `screen_vars` list.

`OptimalPICutoff` *Optimal Cutoff for Prognostic Index on Training Set*

Description

Identifies the optimal cutoff value of a Prognostic Index (PI) to stratify subjects into prognostic groups. It supports `COXNet` and `AFTNet` models with several distributions.

Usage

```
OptimalPICutoff(
  X,
  Y,
  delta,
  beta,
  method = NULL,
  model = NULL,
  dist = NULL,
  probs = seq(0.25, 0.8, by = 0.05),
  plot = FALSE,
  palette = NULL
)
```

Arguments

X	Numeric matrix of covariates.
Y	Numeric vector of observed survival times (log-transformed under AFTNet).
delta	Integer vector of censoring indicators (1 = event, 0 = censored).
beta	Numeric vector of estimated regression coefficients obtained from the training set.
method	Character string specifying the cutoff selection method ("median" or "minpvalue").
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet").
dist	Character string specifying the AFTNet distribution. Must be one of "weibull", "lognormal", or "loglogistic".
probs	Vector of probabilities used when method = "minpvalue" to generate candidate cutoffs based on quantiles of the PI (default: probs = seq(0.25, 0.80, by = 0.05)).
plot	Logical value indicating whether survival curves should be produced (default: FALSE).
palette	Optional character vector of length 2 specifying colors used for the survival curves. For "COXNet", colors correspond to high- and low-risk groups. For "AFTNet", colors correspond to short- and long-survival groups. If NULL, default colors are used.

Details

The Prognostic Index (PI) is computed as a linear predictor. Two alternative strategies are available to define the cutoff.

- **Median-based cutoff**: Subjects are dichotomized as follows:
  - COXNet: PI $\geq$ median is *High Risk*, otherwise *Low Risk*.
  - AFTNet: - PI $\geq$ median is *Short Survival*, otherwise *Long Survival*.
- **Minimum p-value approach**: A grid of candidate cutoffs is generated from the quantiles of the PI. For each candidate:
  - The cohort is dichotomized according to the model-specific direction.
  - Two models are fitted (full model including the group indicator, and null model without the group indicator).
  - A likelihood ratio (LR) test is performed between the two models.

Model fitting is performed using survival::coxph() for COXNet, or survival::survreg() for AFTNet.

The raw p-values are adjusted for multiple testing using the Benjamini–Hochberg procedure. The optimal cutoff corresponds to the smallest adjusted p-value.

If plot = TRUE, survival curves are generated (Kaplan–Meier curves for COXNet, parametric survival curves based on the selected distribution for AFTNet).

Value

For `method = "median"`, a list with

- `cutoff`: numeric cutoff value.
- `PI.data`: data frame containing the PI, survival time, status, and group labels.

For `method = "minpvalue"`, the list additionally contains:

- `summary`: table of p-values across candidate quantiles.
- `optimal`: optimal cutoff information (quantile, cutoff value, raw and adjusted p-values).

`PathwayDashboard` *Interactive Pathway Analysis Dashboard*

Description

Constructs interactive pathway analysis networks and generates an HTML dashboard from a list of genes. Pathways can be retrieved via KEGG database or provided through a custom file.

Usage

```
PathwayDashboard(
  genes_list,
  header = TRUE,
  useKeggAPI = TRUE,
  pathway_file = NULL,
  nodesCols = c("#5C7997", "#F5C59F"),
  diseaseNodes = FALSE,
  disease_file = NULL,
  top_percent = 20,
  batch_size = 10,
  background_genes = NULL,
  min_genes = 2,
  top_n = 10,
  db_name = "org.Hs.eg.db",
  organism = "hsa",
  out_dir = NULL,
  open_browser = TRUE,
  verbose = FALSE
)
```

Arguments

`genes_list`	Character vector of gene symbols, a file path to a tab-delimited file, or a data frame where the first column contains gene symbols.
`header`	Logical value indicating whether the input file has a header (`default: TRUE`).

useKeggAPI	Logical value indicating whether to use the KEGG REST API to retrieve pathways (default: TRUE).
pathway_file	Optional data frame or file path containing custom pathway data. Required if useKeggAPI = FALSE. Must have columns: pathway, gene, optional name.
nodesCols	Character vector of length 2 defining node colors. First color for regular nodes, second for highlighted nodes (when diseaseNodes = TRUE).
diseaseNodes	Logical value indicating whether to highlight disease-associated nodes (default: TRUE).
disease_file	Optional file path or data frame containing disease-associated gene scores. Must have at least two columns: gene and score.
top_percent	Numeric value indicating the percentage of top genes to highlight based on disease_file (used with diseaseNodes, default: 20).
batch_size	Numeric value indicating the batch size for KEGG API queries (default: 10).
background_genes	Optional vector of background genes for enrichment analysis.
min_genes	Numeric value indicating minimum number of genes in a pathway to be considered (default: 2).
top_n	Numeric value indicating the number of top pathways to display in the dashboard (default: 10).
db_name	Character string specifying the Bioconductor Annotation DB name for gene mapping (default: "org.Hs.eg.db").
organism	Character string specifying KEGG organism code (default: "hsa").
out_dir	Character string specifying output directory for results.
open_browser	Logical value; if TRUE and interactive session, opens dashboard in browser (default: TRUE).
verbose	Logical value, if TRUE progress messages are printed.

Details

Workflow implemented by the function:

1. Converts gene symbols to Entrez IDs for KEGG queries and maps back to gene symbols after pathway retrieval.
2. Retrieves pathways using KEGG API if useKeggAPI = TRUE, otherwise uses pathway_file.
3. Constructs a gene-pathway binary incidence matrix (genes as rows, pathways as columns).
4. Builds an igraph network where genes are nodes and edges link genes in the same pathways.
5. Assigns node colors based on connectivity and optional disease association.
6. Highlights top genes by connectivity or disease association using nodesCols and top_percent.
7. Saves network information in network_data.rds and optionally renders an interactive HTML dashboard (Dashboard.html).

The network_data.rds object contains:

- g: igraph object representing the network.

- edge_info: data frame with edges, colors, and pathway labels.
- legend_info: legend codes, colors, and counts for pathways.
- all_genes, conn_genes: all input genes and connected genes.
- node_colours: node colors and borders for plotting.
- pathway_df: data frame of pathways and genes.
- background, min_genes, top_n: parameters.

Value

Saves:

- network_data.rds: serialized network object for later use.
- Dashboard.html: interactive dashboard showing network and enrichment panels.

Note

If useKeggAPI = TRUE, the function queries the KEGG REST API to retrieve pathway information. An active internet connection is required in this case. Moreover, gene names conversion relies on local Bioconductor Annotation DBs (e.g., org.Hs.eg.db). The function returns paths to generated files but does not print to console or open files unless explicitly requested.

See Also

Enrichment for pathway enrichment results.

PlotCvNet *Plot CV-LP Curve for* COXNet *and* AFTNet

Description

Produces a **ggplot2** visualization of the cross-validation curve obtained from CvNet. The plot displays the CV error as a function of $\log(\lambda)$ with optional error bars, and reference lines for lambda.min and lambda.1se.

Usage

```
PlotCvNet(cv.out, alpha = NULL, errorbar = FALSE, colors = NULL)
```

Arguments

cv.out — Object of class "cv.out" (returned by CvNet), containing at least:

- cv.err.linPred: mean CV errors for linear predictor.
- lambda.grid: grid of λ values used as regularization path.
- lambda.min: value of λ minimizing the CV error.
- lambda.1se: largest λ within one standard error.

- `cvup`: upper error curve.
- `cvlo`: lower error curve.

`alpha` Numeric parameter controlling the convex combination of the two penalty terms (value in `[0,1]`), used only for plot annotation (`default: NULL`).

`errorbar` Logical value, if `TRUE` the plot includes vertical error bars representing 1se of the cross-validation error at each fold (`default: FALSE`).

`colors` Optional named list of colors:

- `line`: color of the cross-validation error curve.
- `points`: color of observed CV error evaluations.
- `min`: color of the vertical line indicating `lambda.min`.
- `one_se`: color of the vertical line indicating `lambda.1se`.

If `NULL`, a default color palette is used.

Value

A **ggplot2** object showing the CV-LP curve.

`ProxGDNet` *Proximal Gradient Descent for* `COXNet` *and* `AFTNet`

Description

Estimate the regression coefficients in `COXNet` and `AFTNet` models using a proximal gradient descent algorithm. The objective function combines the normalized negative (partial) log-likelihood with an ℓ_1 penalty, and a Laplacian regularization term.

Usage

```
ProxGDNet(
  X,
  Y,
  delta,
  L = NULL,
  beta0,
  alpha,
  lambda,
  model = NULL,
  dist = NULL,
  sigma = NULL,
  value = 2,
  niter = 1000,
  conv = 0.001
)
```

Arguments

X	Numeric matrix of standardized covariates.
Y	Numeric vector of observed survival times (log-transformed under AFTNet).
delta	Integer vector of censoring indicators (1 = event, 0 = censored).
L	Optional positive semi-definite, symmetric, and diagonally dominant Laplacian matrix encoding prior network information (see CreateNetwork for details). If NULL, no network-based penalization is applied.
beta0	Numeric vector of initial regression coefficients.
alpha	Numeric parameter controlling the convex combination of the two penalty terms (value in [0,1]).
lambda	Non-negative regularization parameter.
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet").
dist	Character string specifying the error distribution in AFTNet model. Must be one of "weibull", "lognormal", or "loglogistic".
sigma	Positive numeric scalar representing the scale parameter of the error distribution in AFTNet model.
value	Numeric scalar greater than 1 specifying the multiplicative factor used to increase the step-size constant during backtracking line search (default: 2).
niter	Maximum number of iterations (default: 1000).
conv	Convergence tolerance (default: 1e-3).

Details

The algorithm minimizes the objective function:

$$\mathcal{L}(\beta) = -\frac{1}{n}\ell(\beta) + \lambda\alpha\|\beta\|_1 + \lambda(1-\alpha)\beta^\top \mathbf{L}\beta$$

where $\ell(\beta)$ is the log-likelihood (partial for COXNet), $\|\beta\|_1$ is the LASSO penalty, $\beta^\top \mathbf{L}\beta$ is the Laplacian constraint.

At each iteration the method performs the backtracking line search to enforce a sufficient decrease condition, the gradient step size adaptation (initialized as Lipschitz constant), and an early stopping based on relative change in objective function.

Convergence is reached when either the maximum number of iterations is attained, or the relative change in the objective function between consecutive iterations falls below the specific tolerance conv.

Value

A list with the following components

- beta: numeric vector of estimated regression coefficients.
- objective: numeric scalar, the final value of the objective function.
- iterations: number of iterations performed until convergence (or until the maximum number of iterations niter is reached).

RepositoryDisease *Disease-Specific Gene Repository from HumanBase*

Description

Download disease-associated gene predictions from the HumanBase resource. The function retrieves gene-level association scores for a given Disease Ontology ID (DOID) and returns a tidy data frame containing gene identifiers and scores.

Usage

```
RepositoryDisease(
  doid = NULL,
  cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

doid	Character string specifying Disease Ontology ID ("DOID:XXXX").
cache	Logical value; if TRUE, downloaded HumanBase files are cached for reuse in cache_dir. If FALSE (default), files are downloaded for the current session only.
cache_dir	Character string specifying a directory used to cache downloaded HumanBase files (when cache = TRUE).
verbose	Logical value, if TRUE progress messages are printed.

Value

A data frame with three columns:

- entrez_id: Entrez gene identifier.
- standard_name: Gene symbol.
- score: Association score from HumanBase.

Note

An active internet connection is required.

Examples

```
# - Download disease-specific gene repository for Lung Adenocarcinoma -

   disease_genes <- RepositoryDisease(
    doid      = "DOID:1324",
```

```
 cache     = FALSE,
 cache_dir = NULL,
 verbose   = FALSE
)$standard_name

head(disease_genes)
```

`RepositoryTissue` *Tissue-Specific Top Edge Network from HumanBase*

Description

Downloads the top edge gene interaction network for a specific human tissue from the HumanBase resource.

Usage

```
RepositoryTissue(
  tissue = NULL,
  cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

`tissue`	Character string specifying the name of the tissue to download. Spaces will automatically be converted to underscores.
`cache`	Logical value; if `TRUE`, downloaded HumanBase files are cached for reuse in `cache_dir`. If `FALSE` (default), files are downloaded for the current session only.
`cache_dir`	Character string specifying a directory used to cache downloaded HumanBase files (when `cache = TRUE`).
`verbose`	Logical value, if `TRUE` progress messages are printed.

Value

A data.frame with tissue-specific gene interactions (columns: `gene1`, `gene2`, and `score`).

Note

An active internet connection is required.

Examples

```
# - Download tissue-specific repository for Lung Adenocarcinoma -

   tissue <- RepositoryTissue(
    tissue    = "lung",
    cache     = FALSE,
    cache_dir = NULL,
    verbose   = FALSE
   )

   head(tissue)
```

`Simulations` *Simulate Transcription Factor (TF) Target Gene Networks with Survival Outcomes*

Description

Generates structured gene expression data based on TFs and their regulated target genes, together with survival outcomes simulated from `COXNet` or `AFTNet` models. The function supports both **independent** and **interconnected** TF modules with user-defined shared targets via `shared_scheme`.

Usage

```
Simulations(
  n,
  r,
  targets,
  p_active,
  rho = 0.7,
  rate = 0.5,
  b_true = c(0.8, 1.2, -1.2, -0.8),
  nsimul = 10,
  model = NULL,
  baseline = NULL,
  phi = 0.1,
  sigma_true = 1,
  breaks = c(0, 6, 36, 60),
  hazards = c(0.15, 0.005, 0.1),
  shared_scheme = NULL,
  choice = 1,
  save = FALSE,
  save_path = NULL,
  seed = 2026,
  verbose = FALSE
)
```

Arguments

n	Numeric value of observations.
r	Numeric value of TFs (for interconnected modules, at least 4 TFs are recommended).
targets	Numeric value of target genes regulated by each TF.
p_active	Numeric value of truly active predictors (non-zero coefficients).
rho	Numeric value of correlation between each TF and its target (default: 0.70).
rate	Numeric value of desired censoring proportion (default: 0.50).
b_true	Numeric vector of length 4 (pos_min, pos_max, neg_min, neg_max) used to generate positive and negative non-zero coefficients.
nsimul	Numeric value of simulated datasets (default: 10).
model	Character string specifying the survival model used for simulation ("COXNet", or "AFTNet").
baseline	Character string specifying baseline hazard distribution. • For COXNet: exponential ("exp"), Weibull ("weibull"), or piecewise-constant ("piecewise"). • For AFTNet: Weibull ("weibull"), Log-Normal ("lognormal"), or Log-Logistic ("loglogistic").
phi	Numeric value of frailty parameter for COXNet's baselines (required for "exp" and "weibull").
sigma_true	Positive numeric scalar representing the scale parameter of the error distribution in AFTNet model (default: 1).
breaks	Numeric vector of time breakpoints for piecewise exponential hazards (required if baseline = "piecewise", default: c(0, 6, 36, 60)).
hazards	Numeric vector of hazard rates corresponding to each interval in breaks (default: c(0.15, 0.005, 0.1)).
shared_scheme	List defining interconnected TF modules. If NULL (default), TFs regulate disjoint target sets (independent structure). Otherwise, it must be a list of modules, each containing • tfs: integer vector of TF indices in the module, • shared: number of genes shared among those TFs, • unique: integer vector giving the number of TF-specific targets.
choice	Value specifying the choice for the signs of the adjacency matrix • 1 (default): for correlation-based signs, • 2: for ridge-based signs (see CreateNetwork for details).
save	Logical value, if TRUE each simulated dataset is saved as an .rds file in the directory specified by save_path (default: FALSE).
save_path	Character string specifying an existing directory used only when save = TRUE. No files are written by default.
seed	Random seed for reproducibility (default: 2026).
verbose	Logical value, if TRUE progress and summary messages are printed during simulation (default: FALSE).

Details

The total number of predictors is given by $p = r \times (targets + 1)$, where each TF contributes one regulatory variable in addition to its associated target genes.

The function supports two alternative network topologies

- **Independent structure**: each TF regulates its own targets independently.
- **Interconnected structure**: TFs specified in the same `shared_scheme` share `shared` genes and additionally have their own unique genes as specified in `unique`.

These regulatory relationships are encoded in the adjacency matrix, which exhibits a block-diagonal structure under independence, and introduces cross-connections between TFs and shared targets when modules are specified.

Survival times are generated according to the chosen baseline distribution and linear predictors derived from the simulated gene expression data. Optional frailty effects and censoring are included, with the censoring mechanism calibrated to achieve the desired censoring proportion specified by `rate`.

The function also returns the true regression coefficients, allowing the user to evaluate variable selection performance using measures such as false positive and false negative rates.

Value

A list with the following components:

- `X_list`: list of simulated design matrices.
- `beta_list`: list of true regression coefficient vectors.
- `time_list`: list of observed survival times (log-transformed under `AFTNet`).
- `delta_list`: list of censoring indicators (1 = event, 0 = censored).
- `L_list`: list of Laplacian matrices representing the TF–gene regulatory network.

Examples

```
# - Simulate interconnected structure under Weibull-COXNet model -

  targets <- 10
  s1 <- 5
  s2 <- 3

  shared_scheme <- list(
  list(tfs = c(1, 3), shared = s1, unique = c(targets - s1, targets - s1)),
  list(tfs = c(2, 4), shared = s2, unique = c(targets - s2, targets - s2)))

  simul_data <- Simulations(
  n = 165, r = 40, targets = targets, p_active = 40,
  b_true = c(0.8,1.2,-1.2,-0.8),
  rate = 0.3, nsimul = 1,
  model = "COXNet", baseline = "weibull",
  shared_scheme = shared_scheme,
  seed = 2026, verbose = FALSE)
```

```
# Extract the Laplacian matrix

  L <- simul_data$L[[1]]

# This matrix uncovers the topological overlap between TFs:
# TF1 and TF3 co-regulate 5 genes, while TF2 and TF4 share 3 target genes.
```

`ValidationPI` *Prognostic Index Validation on Testing Set*

Description

Validates a Prognostic Index (PI) obtained from a fitted survival model (`COXNet` or `AFTNet`) on an independent testing set. Given the estimated regression coefficients, it computes the PI for each subject, assigns prognostic groups using a pre-specified optimal cutoff, and evaluates survival separation and statistical significance.

Usage

```
ValidationPI(
  X,
  Y,
  delta,
  beta,
  opt_cutoff,
  model = NULL,
  dist = NULL,
  plot = FALSE,
  palette = NULL
)
```

Arguments

`X`	Numeric matrix of testing covariates scaled using the training data.
`Y`	Numeric vector of observed testing survival times (log-transformed under `AFTNet`).
`delta`	Integer vector of testing censoring indicators (1 = event, 0 = censored).
`beta`	Numeric vector of estimated regression coefficients obtained from the training set.
`opt_cutoff`	Numeric cutoff value used to split the PI into two prognostic groups.
`model`	Character string specifying the fitted survival model (`"COXNet"`, or `"AFTNet"`).
`dist`	Character string specifying the `AFTNet` distribution. Must be one of `"weibull"`, `"lognormal"`, or `"loglogistic"`.
`plot`	Logical value, if `TRUE` returns the combined survival plot (`default: FALSE`).

palette — Optional character vector of length 2 specifying colors used for the survival curves. For "COXNet", colors correspond to high- and low-risk groups. For "AFTNet", colors correspond to long- and short-survival groups. If NULL, default colors are used.

Details

For COXNet, Kaplan-Meier survival curves are computed, a log-rank test is performed, and the $PI = X\beta$ is compared to opt_cutoff to define *High Risk* and *Low Risk* groups.

For AFTNet, parametric survival curves are computed using the specified distribution, a likelihood ratio test is performed, and the $PI = -X\beta$ is compared to opt_cutoff to define *Short Survival* and *Long Survival* groups.

The function also produces:

- Survival curves with group-specific colors,
- Risk tables (number-at-risk) aligned with survival curves,
- Distribution plots of the PI across groups.

Value

A list containing:

- df: data frame with columns PI (prognostic index for each subject), Y, delta, groupRisk (assigned prognostic group based on opt_cutoff),
- p_value: from the log-rank test (COXNet) or likelihood ratio test (AFTNet), measuring survival separation between groups.

See Also

OptimalPICutoff for opt_cutoff value selection.

VariableScreening — *Variables Screening Methods Based on Prior Knowledge and Marginal Utility*

Description

Reduces the high-dimensional feature space to a more manageable subset of variables by applying one of three screening strategies:

- **BMD (Biomedical-driven)**: selects covariates based on prior biomedical knowledge about their relevance to the disease under investigation,
- **DAD (Data-driven)**: selects features using component-wise estimators obtained from the chosen penalized model,
- **BMD+DAD**: combines both biomedical knowledge and data-driven insights.

Usage

```
VariableScreening(
  X,
  Y,
  delta,
  disease_genes,
  screening = NULL,
  model = NULL,
  dist = NULL,
  rank_method = NULL,
  d = NULL,
  standardize = TRUE,
  verbose = FALSE
)
```

Arguments

X	Numeric matrix of covariates.
Y	Numeric vector of observed survival times (log-transformed under AFTNet).
delta	Integer vector of censoring indicators (1 = event, 0 = censored).
disease_genes	Character vector containing the names of genes known to be associated with diseases.
screening	Character string specifying the screening method ("BMD", "DAD", or "BMD+DAD").
model	Character string specifying the fitted survival model ("COXNet", or "AFTNet") required for DAD-based screening.
dist	Character string specifying the AFTNet distribution. Must be one of "weibull", "lognormal", or "loglogistic".
rank_method	Character string specifying the ranking criterion for DAD-based screening: "absmg" (absolute marginal coefficients), "mg" (marginal function), or "mgpadj" (adjusted p-value from the marginal function).
d	Numeric value representing the threshold for top-ranked features to select in DAD-based screening (default: NULL).
standardize	Logical value indicating whether to standardize the input matrix in DAD-based screening: • if TRUE (default) each column is centered to have mean 0 and scaled to have unit variance. • if FALSE the function assumes that the matrix has already been standardized by the user.
verbose	Logical value, if TRUE progress messages are printed (default: FALSE).

Details

The function uses marginal ranking approaches to select features based on their association with survival outcomes.

- In the **BMD** approach, prior knowledge comes from literature or external biological databases such as HumanBase.
- The **DAD** screening computes marginal regression coefficients to rank features according to their estimated importance under the selected model:
  - `absmg`: top `d` covariates by largest absolute marginal coefficients.
  - `mg`: top `d` covariates by largest marginal coefficients, preserving the direction.
  - `mgpadj`: top `d` covariates passing significance thresholds based on adjusted p-values.
- The **BMD+DAD** combines prior biological knowledge and data-driven selection for comprehensive feature screening.

Value

A list containing selected variable names `screen_vars`.

See Also

`CreateNetwork` or `RepositoryDisease` for the `disease_genes` names.

Index