

Package: inferCSN (via r-universe)

October 14, 2026

Type Package

Title Inferring Cell-Specific Gene Regulatory Network

Version 1.2.5

Date 2026-09-06

Maintainer Meng Xu <mengxu98@qq.com>

Description Inferring cell-type specific gene regulatory network using sparse regression model from single-cell transcriptomic data.

License MIT + file LICENSE

URL <https://mengxu98.github.io/inferCSN/>

BugReports <https://github.com/mengxu98/inferCSN/issues>

Depends R (>= 4.1.0)

Imports cli, Matrix, methods, Rcpp, stats, thisutils

Suggests LOLearn, testthat

LinkingTo Rcpp

Remotes mengxu98/thisutils

Config/Needs/website mengxu98/thistemplate

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Language en-US

Config/roxygen2/version 8.1.0

Repository <https://mengxu98.r-universe.dev>

Date/Publication 2026-10-14 13:33:09 UTC

RemoteUrl <https://github.com/mengxu98/infercsn>

RemoteRef HEAD

RemoteSha 71a73c6e52b770c268a9c947c9112375a84cee84

Contents

inferCSN-package	2
example_ground_truth	2
example_matrix	3
example_meta_data	3
filter_sort_matrix	3
fit_greedy_l0	4
fit_greedy_l0_batch	4
inferCSN	5
infercsn_logo	7
network_format	8
print.infercsn_logo	8
select_trend_features	9
single_network	10
Index	12

inferCSN-package	<i>Inferring Cell-Specific Gene Regulatory Network</i>
------------------	--

Description

Inferring cell-type specific gene regulatory network using sparse regression model from single-cell transcriptomic data.

Author(s)

Meng Xu (Maintainer), <mengxu98@qq.com>

Source

<https://mengxu98.github.io/inferCSN/>

See Also

- Useful links:
- <https://mengxu98.github.io/inferCSN/>
 - Report bugs at <https://github.com/mengxu98/inferCSN/issues>

example_ground_truth	<i>Example ground truth data</i>
----------------------	----------------------------------

Description

Ground-truth edges for the bundled synthetic example.

example_matrix	<i>Example matrix data</i>
----------------	----------------------------

Description

Expression matrix for the bundled synthetic example.

example_meta_data	<i>Example meta data</i>
-------------------	--------------------------

Description

Cell metadata for the bundled synthetic example.

filter_sort_matrix	<i>Filter and sort a network matrix</i>
--------------------	---

Description

Filter and sort a network matrix

Usage

```
filter_sort_matrix(network_matrix, regulators = NULL, targets = NULL)
```

Arguments

network_matrix Network weight matrix.
regulators, targets Nodes to include.

Value

A filtered and sorted matrix.

fit_greedy_l0	<i>Fit greedy L0 regression</i>
---------------	---------------------------------

Description

Fits a standardized sparse linear model by BIC-decreasing support updates.

Usage

```
fit_greedy_l0(
  x,
  y,
  max_support_size = NULL,
  min_improvement = 1e-10,
  verbose = TRUE
)
```

Arguments

x	Numeric predictor matrix with observations in rows.
y	Numeric response vector.
max_support_size	Optional positive integer support limit.
min_improvement	Minimum RSS improvement considered.
verbose	Logical verbosity flag.

Value

A list with 'model', 'metrics', and 'coefficients'.

fit_greedy_l0_batch	<i>Fit batched greedy L0 regressions</i>
---------------------	--

Description

Fits multiple responses from shared predictor cross-products.

Usage

```
fit_greedy_l0_batch(
  gram,
  xty,
  response_ss,
  candidates,
  n_obs,
  max_support_size = NULL,
  min_improvement = 1e-10
)
```

Arguments

gram	Numeric square predictor cross-product matrix.
xty	Numeric predictor-by-response cross-product matrix.
response_ss	Numeric response sums of squares.
candidates	List of one-based predictor indices, one element per response.
n_obs	Number of observations used for the cross-products.
max_support_size	Optional positive integer support limit.
min_improvement	Minimum RSS reduction proposed during forward search.

Value

Batched coefficients, deletion evidence, and fit summaries.

inferCSN	<i>inferring cell-type specific gene regulatory network</i>
----------	---

Description

inferring cell-type specific gene regulatory network

Usage

```
inferCSN(
  object,
  pseudotime = NULL,
  regulators = NULL,
  targets = NULL,
  max_support_size = NULL,
  lag_fraction = 0.05,
  lag_steps = NULL,
  cores = 1,
```

```

    verbose = TRUE,
    method = c("greedy_l0", "L0", "L0L1", "L0L2"),
    ...
)

## S4 method for signature 'matrix'
inferCSN(
  object,
  pseudotime = NULL,
  regulators = NULL,
  targets = NULL,
  max_support_size = NULL,
  lag_fraction = 0.05,
  lag_steps = NULL,
  cores = 1,
  verbose = TRUE,
  method = c("greedy_l0", "L0", "L0L1", "L0L2"),
  ...
)

## S4 method for signature 'sparseMatrix'
inferCSN(
  object,
  pseudotime = NULL,
  regulators = NULL,
  targets = NULL,
  max_support_size = NULL,
  lag_fraction = 0.05,
  lag_steps = NULL,
  cores = 1,
  verbose = TRUE,
  method = c("greedy_l0", "L0", "L0L1", "L0L2"),
  ...
)

```

Arguments

<code>object</code>	Numeric expression matrix with cells in rows and genes in columns.
<code>pseudotime</code>	Optional pseudotime vector or branch matrix for either method. Regulators in earlier states predict targets in later states; tied states are averaged and shared branch transitions are counted once.
<code>regulators, targets</code>	Optional gene subsets.
<code>max_support_size</code>	Optional support-size limit.
<code>lag_fraction</code>	Fractional lag used when <code>lag_steps</code> is <code>NULL</code> .
<code>lag_steps</code>	Optional integer lag. LOLearn centers and scales lagged expression within each branch and defaults to fitting without an intercept.

cores	Number of inference workers.
verbose	Whether to report progress.
method	greedy_l0 (default), or LOLearn with the L0, L0L1, or L0L2 penalty.
...	Arguments passed to the method.

Value

A data frame containing exactly regulator, target, and weight.

Examples

```
data(example_matrix)
data(example_meta_data)
network_table <- inferCSN(
  example_matrix,
  pseudotime = example_meta_data$pseudotime
)
head(network_table)

inferCSN(
  example_matrix,
  regulators = c("g1", "g2"),
  targets = c("g3", "g4")
)
```

infercsn_logo	<i>The logo of inferCSN</i>
---------------	-----------------------------

Description

The inferCSN logo, using ASCII or Unicode characters Use [cli::ansi_strip](#) to get rid of the colors.

Usage

```
infercsn_logo(unicode = cli::is_utf8_output())
```

Arguments

unicode	Unicode symbols on UTF-8 platforms. Default is cli::is_utf8_output .
---------	--

Value

A character vector.

References

<https://github.com/tidyverse/tidyverse/blob/main/R/logo.R>

Examples

```
infercsn_logo()
```

network_format	<i>Format a network table</i>
----------------	-------------------------------

Description

Format a network table

Usage

```
network_format(  
  network_table,  
  regulators = NULL,  
  targets = NULL,  
  abs_weight = TRUE  
)
```

Arguments

- network_table Network edge table.
- regulators, targets Nodes to include.
- abs_weight Whether to use absolute weights and add interaction signs.

Value

A formatted network edge table.

print.infercsn_logo	<i>Print logo</i>
---------------------	-------------------

Description

Print logo

Usage

```
## S3 method for class 'infercsn_logo'  
print(x, ...)
```

Arguments

- x Input information.
- ... Other parameters.

Value

Print the ASCII logo

select_trend_features *Select features with significant trends*

Description

Select features with significant trends

Usage

```
select_trend_features(
  x = NULL,
  pseudotime = NULL,
  method = c("pretsa", "gam"),
  padjust_threshold = 0.05,
  n_candidates = NULL,
  statistics = NULL,
  ...
)
```

Arguments

x	Numeric matrix with observations in rows and features in columns, or NULL when supplying statistics.
pseudotime	Numeric coordinate per observation.
method	Fitting method: "pretsa" or "gam".
padjust_threshold	Strict adjusted P-value cutoff between zero and one.
n_candidates	Maximum number of selected features, or NULL for no cap.
statistics	Precomputed table with feature row names and padjust (or pvalue if padjust is absent).
...	Numerical options passed to thisutils::fit_trends() .

Value

A list with features, statistics, and fit (NULL for precomputed input). Nonfinite P-values are excluded. Input order is retained unless the cap is exceeded; then the smallest P-values are selected with stable ties.

Examples

```
t <- seq(0, 1, length.out = 40)
x <- cbind(a = sin(t * 5), b = cos(t * 3))
select_trend_features(x, t)$features
```

single_network	<i>Construct network for single target gene</i>
----------------	---

Description

Construct network for single target gene

Usage

```
single_network(
  matrix,
  regulators,
  target,
  pseudotime = NULL,
  max_support_size = NULL,
  lag_fraction = 0.05,
  lag_steps = NULL,
  cores = 1,
  verbose = TRUE,
  method = c("greedy_l0", "L0", "L0L1", "L0L2"),
  ...
)
```

Arguments

matrix	An expression matrix.
regulators	Candidate regulator genes.
target	The target gene.
pseudotime	Optional pseudotime vector or branch matrix passed to [inferCSN()].
max_support_size	Optional support-size cap passed to [inferCSN()].
lag_fraction	Fractional state lag passed to [inferCSN()].
lag_steps	Optional integer state lag passed to [inferCSN()].
cores	Number of inference workers.
verbose	Whether to report progress.
method	greedy_l0 (default), or L0Learn with the L0, L0L1, or L0L2 penalty.
...	Arguments passed to the method.

Value

A data frame with regulator, target, and weight columns. Greedy-L0 returns selected edges. L0Learn retains the original per-regulator coefficients, including zeros; [inferCSN()] removes zero weights from the complete network.

Examples

```
data(example_matrix)
head(
  single_network(
    example_matrix,
    regulators = colnames(example_matrix),
    target = "g1"
  )
)
single_network(
  example_matrix,
  regulators = c("g1", "g2", "g3"),
  target = "g1"
)
```

Index

`cli::ansi_strip`, [7](#)
`cli::is_utf8_output`, [7](#)

`example_ground_truth`, [2](#)
`example_matrix`, [3](#)
`example_meta_data`, [3](#)

`filter_sort_matrix`, [3](#)
`fit_greedy_l0`, [4](#)
`fit_greedy_l0_batch`, [4](#)

`inferCSN`, [5](#)
`inferCSN, matrix-method` (`inferCSN`), [5](#)
`inferCSN, sparseMatrix-method`
 (`inferCSN`), [5](#)
`inferCSN-package`, [2](#)
`infercsn_logo`, [7](#)

`network_format`, [8](#)

`print.infercsn_logo`, [8](#)

`select_trend_features`, [9](#)
`single_network`, [10](#)

`thisutils::fit_trends()`, [9](#)