Package ‘magrene’

March 14, 2024

Title  Motif Analysis In Gene Regulatory Networks
Version  1.4.0
Date  2022-01-21

Description  magrene allows the identification and analysis of graph motifs in (duplicated) gene regulatory networks (GRNs), including lambda, V, PPI V, delta, and bifan motifs. GRNs can be tested for motif enrichment by comparing motif frequencies to a null distribution generated from degree-preserving simulated GRNs. Motif frequencies can be analyzed in the context of gene duplications to explore the impact of small-scale and whole-genome duplications on gene regulatory networks. Finally, users can calculate interaction similarity for gene pairs based on the Sorensen-Dice similarity index.

License  GPL-3

URL  https://github.com/almeidasilvaf/magrene

BugReports  https://support.bioconductor.org/t/magrene

biocViews  Software, MotifDiscovery, NetworkEnrichment, SystemsBiology, GraphAndNetwork

Encoding  UTF-8

Roxygen  list(markdown = TRUE)

RoxygenNote  7.2.0

Imports  utils, stats, BiocParallel

Suggests  BiocStyle, covr, knitr, rmarkdown, ggplot2, sessioninfo, testthat (>= 3.0.0)

Config/testthat/edition  3

VignetteBuilder  knitr

Depends  R (>= 4.2.0)

LazyData  false

git_url  https://git.bioconductor.org/packages/magrene

git_branch  RELEASE_3_18

git_last_commit  268c106
calculate_Z

Description
Calculate Z-score for motif frequencies

Usage
calculate_Z( observed = NULL, nulls = NULL )

Arguments

observed
A list of observed motif frequencies for each motif type. List elements must be named "lambda", "bifan", "V", "PPI_V", and "delta" (not necessarily in that order).

nulls
A list of null distributions for each motif type as returned by generate_nulls.

Value
A numeric vector with the Z-score for each motif type.
**Examples**

```r
# Simulating it for test purposes
null <- rnorm(1000, mean = 5, sd = 1)
nulls <- list(
  lambda = null, V = null, PPI_V = null, delta = null, bifan = null
)
observed <- list(lambda = 7, bifan = 13, delta = 9, V = 5, PPI_V = 10)
z <- calculate_Z(observed, nulls)
# Check for motif enrichment (Z > 5)
z[which(z > 5)]
```

---

**find_bifan**  
*Find bifan motifs*

**Description**

Find bifan motifs

**Usage**

```r
find_bifan(
  edgelist = NULL,
  paralogs = NULL,
  lambda_vec = NULL,
  count_only = FALSE
)
```

**Arguments**

- `edgelist`  
  A 2-column data frame with regulators in column 1 and targets in column 2. It can be ignored if you give lambda motifs to parameter `lambda_vec` (recommended).

- `paralogs`  
  A 2-column data frame with gene IDs for each paralog in the paralog pair.

- `lambda_vec`  
  A character of lambda motifs as returned by `find_lambda()`. If this is NULL, this function will find lambda motifs from `edgelist` and `paralogs` first. Passing previously identified lambda motifs will make this function much faster.

- `count_only`  
  Logical indicating whether the function should return only motif counts as a numeric scalar. If FALSE, it will return a character vector of motifs. Default: FALSE.

**Value**

A character vector with bifan motifs represented in the format `regulator1, regulator2->target1, target2`. 
find_delta

Examples

data(gma_grn)
data(gma_paralogs)
edgelist <- gma_grn[1:50000, 1:2]
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
paralogs <- rbind(paralogs, data.frame(duplicate1 = "Glyma.01G177200",
                                            duplicate2 = "Glyma.08G116700")
)
lambda_vec <- find_lambda(edgelist, paralogs)
bifan <- find_bifan(paralogs = paralogs, lambda_vec = lambda_vec)

find_delta

Find delta motifs

Description

Find delta motifs

Usage

find_delta(
edgelist = NULL,
paralogs = NULL,
edgelist_ppi = NULL,
lambda_vec = NULL,
count_only = FALSE
)

Arguments

edgelist A 2-column data frame with regulators in column 1 and targets in column 2. It can be ignored if you give lambda motifs to parameter lambda_vec (recommended).
paralogs A 2-column data frame with gene IDs for each paralog in the paralog pair. It can be ignored if you give lambda motifs to parameter lambda_vec (recommended).
edgelist_ppi A 2-column data frame with IDs of genes that encode each protein in the interacting pair.
lambda_vec A character of lambda motifs as returned by find_lambda(). If this is NULL, this function will find lambda motifs from edgelist and paralogs first. Passing previously identified lambda motifs will make this function much faster.
count_only Logical indicating whether the function should return only motif counts as a numeric scalar. If FALSE, it will return a character vector of motifs. Default: FALSE.
find_lambda

Value
A character vector with lambda motifs represented in the format \texttt{target1<-regulator->target2}.

Examples

data(gma_grn)
data(gma_paralogs)
data(gma_ppi)
edgelist <- gma_grn[500:1000, 1:2] # reducing for test purposes
edgelist <- gma_grn[1:10000, 1:2]
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
edgelist_ppi <- gma_ppi
lambda_vec <- find_lambda(edgelist, paralogs)
motifs <- find_delta(edgelist_ppi = edgelist_ppi, lambda_vec = lambda_vec)

default_lambda

Find lambda motifs

Usage
find_lambda(edgelist = NULL, paralogs = NULL, count_only = FALSE)

Arguments

edgelist A 2-column data frame with regulators in column 1 and targets in column 2.
paralogs A 2-column data frame with gene IDs for each paralog in the paralog pair.
count_only Logical indicating whether the function should return only motif counts as a numeric scalar. If FALSE, it will return a character vector of motifs. Default: FALSE.

Value
A character vector with lambda motifs represented in the format \texttt{target1<-regulator->target2}.

Examples

data(gma_grn)
data(gma_paralogs)
edgelist <- gma_grn[500:1000, 1:2] # reducing for test purposes
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
motifs <- find_lambda(edgelist, paralogs)
find_ppi_v

Find V motifs in protein-protein interactions

Description
Find V motifs in protein-protein interactions

Usage
find_ppi_v(edgelist = NULL, paralogs = NULL, count_only = FALSE)

Arguments
edgelist A 2-column data frame with protein 1 in column 1 and protein 2 in column 2.
paralogs A 2-column data frame with gene IDs for each paralog in the paralog pair.
count_only Logical indicating whether the function should return only motif counts as a numeric scalar. If FALSE, it will return a character vector of motifs. Default: FALSE.

Details
This function aims to find the number of paralogous gene pairs that share an interaction partner.

Value
A character vector with V motifs represented in the format paralog1-partner-paralog2.

Examples
data(gma_ppi)
data(gma_paralogs)
edgelist <- gma_ppi
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
motifs <- find_ppi_v(edgelist, paralogs)

find_v

Find V motifs

Description
Find V motifs

Usage
find_v(edgelist = NULL, paralogs = NULL, count_only = FALSE)
**generate_nulls**

**Arguments**

- **edgelist**
  A 2-column data frame with regulators in column 1 and targets in column 2.

- **paralogs**
  A 2-column data frame with gene IDs for each paralog in the paralog pair.

- **count_only**
  Logical indicating whether the function should return only motif counts as a numeric scalar. If FALSE, it will return a character vector of motifs. Default: FALSE.

**Value**

A character vector with V motifs represented in the format `regulator1->target<-regulator2`.

**Examples**

```r
data(gma_grn)
data(gma_paralogs)
edgelist <- gma_grn[2000:4000, 1:2] # reducing for test purposes
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
motifs <- find_v(edgelist, paralogs)
```

---

**generate_nulls**

Generate null distributions of motif counts for each motif type

**Description**

Generate null distributions of motif counts for each motif type

**Usage**

```r
generate_nulls(
edgelist = NULL,
paralogs = NULL,
edgelist_ppi = NULL,
n = 1000,
bp_param = BiocParallel::SerialParam()
)
```

**Arguments**

- **edgelist**
  A 2-column data frame with regulators in column 1 and targets in column 2.

- **paralogs**
  A 2-column data frame with gene IDs for each paralog in the paralog pair.

- **edgelist_ppi**
  A 2-column data frame with IDs of genes that encode each protein in the interacting pair.

- **n**
  Number of degree-preserving simulated networks to generate. Default: 1000.

- **bp_param**
  BiocParallel back-end to be used. Default: BiocParallel::SerialParam().
Value

A list of numeric vectors named lambda, delta, V, PPI_V, and bifan, containing the null distribution of motif counts for each motif type.

Examples

```r
set.seed(123)
data(gma_grn)
data(gma_paralogs)
data(gma_ppi)
edgelist <- gma_grn[500:1000, 1:2] # reducing for test purposes
paralogs <- gma_paralogs[gma_paralogs$type == "WGD", 1:2]
edgelist_ppi <- gma_ppi
n <- 2 # small n for demonstration purposes
generate_nulls(edgelist, paralogs, edgelist_ppi, n)
```

---

**gma_grn**

*Sample soybean GRN*

**Description**

The GRN was inferred with BioNERO using expression data from Libault et al., 2010, and Severin et al., 2010.

**Usage**

```r
data(gma_grn)
```

**Format**

A 3-column data frame with node1, node2, and edge weight.

**References**


**Examples**

```r
data(gma_grn)
```
**gma_paralogs**

| gma_paralogs | Soybean (Glycine max) duplicated genes |

**Description**

The repertoire of soybean paralogs was retrieved from Almeida-Silva et al., 2020.

**Usage**

```r
data(gma_paralogs)
```

**Format**

A 3-column data frame with duplicate 1, duplicate 2, and duplication type

**References**


**Examples**

```r
data(gma_paralogs)
```

---

**gma_ppi**

| gma_ppi | Sample soybean PPI network |

**Description**

PPI were retrieved from the STRING database and filtered to keep only medium confidence edges and nodes in the GRN.

**Usage**

```r
data(gma_ppi)
```

**Format**

A 2-column data frame with node1 and node2.

**Examples**

```r
data(gma_ppi)
```
nulls

**Null distribution of motif frequencies for vignette data set**

**Description**
Data were filtered exactly as demonstrated in the vignette. Briefly, the top 30k edges from the GRN were kept, and only WGD-derived gene pairs were used.

**Usage**
```r
data(nulls)
```

**Format**
A list of numeric vectors with the motif frequencies in each simulated network. List elements are named *lambda*, *delta*, *V*, *PPI_V*, and *bifan*, and each element has length 100.

**Examples**
```r
data(nulls)
```

sd_similarity

**Calculate Sorensen-Dice similarity between paralogous gene pairs**

**Description**
Calculate Sorensen-Dice similarity between paralogous gene pairs.

**Usage**
```r
sd_similarity(edgelist = NULL, paralogs = NULL)
```

**Arguments**
- **edgelist**: A 2-column data frame with regulators in column 1 and targets in column 2.
- **paralogs**: A 2-column data frame with gene IDs for each paralog in the paralog pair.

**Value**
A data frame containing the paralogous gene pairs and their Sorensen-Dice similarity scores.

**Examples**
```r
data(gma_ppi)
data(gma_paralogs)
edgelist <- gma_ppi
paralogs <- gma_paralogs
sim <- sd_similarity(edgelist, paralogs)
```
Index

* datasets
  gma_grn, 8
  gma_paralogs, 9
  gma_ppi, 9
  nulls, 10

calculate_Z, 2

find_bifan, 3
find_delta, 4
find_lambda, 5
find_ppi_v, 6
find_v, 6

generate_nulls, 7
  gma_grn, 8
  gma_paralogs, 9
  gma_ppi, 9

nulls, 10

sd_similarity, 10