Package ‘cageminer’

May 29, 2024

Title Candidate Gene Miner

Version 1.10.0

Description This package aims to integrate GWAS-derived SNPs and coexpression networks to mine candidate genes associated with a particular phenotype. For that, users must define a set of guide genes, which are known genes involved in the studied phenotype. Additionally, the mined candidates can be given a score that favors candidates that are hubs and/or transcription factors. The scores can then be used to rank and select the top n most promising genes for downstream experiments.

License GPL-3

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

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

biocViews Software, SNP, FunctionalPrediction, GenomeWideAssociation, GeneExpression, NetworkEnrichment, VariantAnnotation, FunctionalGenomics, Network

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports ggplot2, rlang, ggbio, ggtext, GenomeInfoDb, GenomicRanges, IRanges, reshape2, methods, BioNERO

Depends R (>= 4.1)

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

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch RELEASE_3_19

git_last_commit 9facfca

git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-29
Author Fabrício Almeida-Silva [aut, cre]
(https://orcid.org/0000-0002-5314-2964),
Thiago Venancio [aut] (https://orcid.org/0000-0002-2215-8082)
Maintainer Fabrício Almeida-Silva <fabricio_almeidasilva@hotmail.com>

Contents
chr_length .......................................................... 2
gcn ................................................................. 3
gene_ranges .......................................................... 3
 guides ............................................................. 4
 hubs ................................................................. 4
 mine2 ............................................................... 5
 mined_candidates .................................................. 5
 mine_candidates .................................................... 6
 mine_step1 .......................................................... 7
 mine_step2 .......................................................... 8
 mine_step3 .......................................................... 9
 pepper_se .......................................................... 10
 plot_snp_circos ...................................................... 11
 plot_snp_distribution ................................................ 12
 score_genes ........................................................ 12
 simulate_windows ................................................... 13
 snp_pos ............................................................ 14
 tfs ................................................................. 15

Index 16

chr_length  Pepper chromosome lengths

Description
Lengths of pepper chromosomes 1-12 in a GRanges object. The genome for which lengths were calculated (v1.55) was downloaded from http://peppergenome.snu.ac.kr/download.php

Usage
data(chr_length)

Format
A GRanges object
Examples

data(chr_length)

gcn

Simulation of the output list from BioNERO::exp2gcn() with pepper data

Description

This object is a list as returned by BioNERO::exp2gcn(), but only the element genes_and_modules is included. For running time issues, only genes in the cyan module were kept in the element genes_and_modules. All other list elements have been assigned NULL. The network was inferred using the code from the vignette.

Usage

data(gcn)

Format

A list with the elements returned by BioNERO::exp2gcn().

Examples

data(gcn)

gene_ranges

Genomic coordinates of pepper genes

Description


Usage

data(gene_ranges)

Format

A GRanges object

Examples

data(gene_ranges)
**guides**

*Guide genes associated with defense and resistance to oomycetes*

**Description**

The GO annotation was retrieved from PLAZA 4.0 Dicots.

**Usage**

`data(guides)`

**Format**

A data frame with genes in the first column and GO description in the second column.

**References**


**Examples**

`data(guides)`

---

**hubs**

*Example hub genes for the network stored in the gen object*

**Description**

The data frame was created using the code from the vignette.

**Usage**

`data(hubs)`

**Format**

Data frame with gene IDs, module and intramodular degree.

**Examples**

`data(hubs)`
mine2

Example output from mine_step2()

Description
The list was created using the example code from mine_step().

Usage
data(mine2)

Format
List with elements 'candidates' (character vector) and 'enrichment' (data frame).

Examples
data(mine2)

mined_candidates

Example output from mined_candidates()

Description
The data frame was created using the code from the vignette.

Usage
data(mined_candidates)

Format
Data frame with an example of the output from mined_candidates

Examples
data(mined_candidates)
mine_candidates  

Mine high-confidence candidate genes in a single step

Description

Mine high-confidence candidate genes in a single step

Usage

mine_candidates(
  gene_ranges = NULL,
  marker_ranges = NULL,
  window = 2,
  expand_intervals = TRUE,
  gene_col = "ID",
  exp = NULL,
  gcn = NULL,
  guides = NULL,
  metadata,
  metadata_cols = 1,
  sample_group,
  min_cor = 0.2,
  alpha = 0.05,
  ...
)

Arguments

gene_ranges  A GRanges object with genomic coordinates of all genes in the genome.
marker_ranges  Genomic positions of SNPs. For a single trait, a GRanges object. For multiple
traits, a GRangesList or CompressedGRangesList object, with each element of
the list representing SNP positions for a particular trait.
window  Sliding window (in Mb) upstream and downstream relative to each SNP. Default: 2.
expand_intervals  Logical indicating whether or not to expand markers that are represented by
intervals. This is particularly useful if users want to use a custom interval defined
by linkage disequilibrium, for example. Default: TRUE.
gene_col  Column of the GRanges object containing gene ID. Default: "ID", the default
for gff/gff3 files imported with rtracklayer::import.
exp  Expression data frame with genes in row names and samples in column names
or a SummarizedExperiment object.
gcn  Gene coexpression network returned by BioNERO::exp2gcn().
guides  Guide genes as a character vector or as a data frame with genes in the first
column and gene annotation class in the second column.
mine_step1

metadata

Sample metadata with samples in row names and sample information in the first column. Ignored if exp is a SummarizedExperiment object, as the colData will be extracted from the object.

metadata_cols

A vector (either numeric or character) indicating which columns should be extracted from column metadata if exp is a SummarizedExperiment object. The vector can contain column indices (numeric) or column names (character). By default, all columns are used.

sample_group

Level of sample metadata to be used for filtering in gene-trait correlation.

min_cor

Minimum correlation value for BioNERO::gene_significance(). Default: 0.2

alpha

Numeric indicating significance level. Default: 0.05

... Additional arguments to BioNERO::gene_significance.

Value

A data frame with mined candidate genes and their correlation to the condition of interest.

Examples

data(pepper_se)
data(snp_pos)
data(gene_ranges)
data(guides)
data(gcn)
set.seed(1)
candidates <- mine_candidates(gene_ranges, snp_pos, exp = pepper_se,
gcn = gcn, guides = guides$Gene,
sample_group = "PRR_stress")

mine_step1

Step 1: Get all putative candidate genes for a given sliding window

Description

For a user-defined sliding window relative to each SNP, this function will subset all genes whose genomic positions overlap with the sliding window.

Usage

mine_step1(gene_ranges, marker_ranges, window = 2, expand_intervals = TRUE)
Arguments

- **gene_ranges**: A GRanges object with genomic coordinates of all genes in the genome.
- **marker_ranges**: Genomic positions of SNPs. For a single trait, a GRanges object. For multiple traits, a GRangesList or CompressedGRangesList object, with each element of the list representing SNP positions for a particular trait.
- **window**: Sliding window (in Mb) upstream and downstream relative to each SNP. Default: 2.
- **expand_intervals**: Logical indicating whether or not to expand markers that are represented by intervals. This is particularly useful if users want to use a custom interval defined by linkage disequilibrium, for example. Default: TRUE.

Value

A GRanges or GRangesList object with the genomic positions of all putative candidate genes.

See Also

- findOverlaps-methods

Examples

```r
data(snp_pos)
data(gene_ranges)
genes <- mine_step1(gene_ranges, snp_pos, window = 2)
```

---

**mine_step2**  
*Step 2: Get candidates in modules enriched in guide genes*

Description

Step 2: Get candidates in modules enriched in guide genes

Usage

```r
mine_step2(exp, gcn, guides, candidates, ...)
```

Arguments

- **exp**: Expression data frame with genes in row names and samples in column names or a SummarizedExperiment object.
- **gcn**: Gene coexpression network returned by BioNERO::exp2gcn().
- **guides**: Guide genes as a character vector or as a data frame with genes in the first column and gene annotation class in the second column.
- **candidates**: Character vector of all candidates genes to be inspected.
- **...**: Additional arguments to BioNERO::module_enrichment
mine_step3

Value

A list of 2 elements:

- **candidates** Character vector of candidates after step 2
- **enrichment** Data frame of results for enrichment analysis

Examples

```r
data(pepper_se)
data(guides)
data(gcn)
set.seed(1)
mine2 <- mine_step2(
  exp = pepper_se,
  gcn = gcn,
  guides = guides$Gene,
  candidates = rownames(pepper_se)
)
```

mine_step3  

**Step 3: Select candidates based on gene significance**

Description

Step 3: Select candidates based on gene significance

Usage

```r
mine_step3(
  exp,
  metadata,
  metadata_cols = 1,
  candidates,
  sample_group,
  min_cor = 0.2,
  alpha = 0.05,
  ...
)
```

Arguments

- **exp** Expression data frame with genes in row names and samples in column names or a SummarizedExperiment object.
- **metadata** Sample metadata with samples in row names and sample information in the first column. Ignored if `exp` is a SummarizedExperiment object, as the colData will be extracted from the object.
metadata_cols A vector (either numeric or character) indicating which columns should be extracted from column metadata if `exp` is a SummarizedExperiment object. The vector can contain column indices (numeric) or column names (character). By default, all columns are used.

candidates Character vector of candidate genes to be inspected.
sample_group Level of sample metadata to be used for filtering in gene-trait correlation.
min_cor Minimum correlation value for `BioNERO::gene_significance()`. Default: 0.2
alpha Numeric indicating significance level. Default: 0.05
...

Additional arguments to `BioNERO::gene_significance`.

Value
A data frame with mined candidate genes and their correlation to the condition of interest.

Examples
```r
data(pepper_se)
data(snp_pos)
data(gene_ranges)
data(guides)
data(gcn)
data(mine2)
set.seed(1)
mine3 <- mine_step3(
  exp = pepper_se,
  candidates = mine2$candidates,
  sample_group = "PRR_stress"
)
```

---

pepper_se  

*Gene expression data from Kim et al., 2018.*

Description
The data were filtered to keep only the top 4000 genes with highest RPKM values in PRR stress-related samples.

Usage
```r
data(pepper_se)
```

Format
A SummarizedExperiment object.
plot_snp_circos

References


Examples

data(pepper_se)

plot_snp_circos Circos plot of SNP distribution across chromosomes

Description

Circos plot of SNP distribution across chromosomes

Usage

plot_snp_circos(genome_ranges, gene_ranges, marker_ranges)

Arguments

- genome_ranges: A GRanges object with chromosome lengths.
- gene_ranges: A GRanges object with genomic coordinates of all genes in the genome.
- marker_ranges: Genomic positions of SNPs. For a single trait, a GRanges object. For multiple traits, a GRangesList or CompressedGRangesList object, with each element of the list representing SNP positions for a particular trait.

Value

A ggplot object with a circos plot of molecular marker distribution across chromosomes.

Examples

data(snp_pos)
data(gene_ranges)
data(chr_length)
p <- plot_snp_circos(chr_length, gene_ranges, snp_pos)
plot_snp_distribution  
*Plot a barplot of SNP distribution across chromosomes*

**Description**
Plot a barplot of SNP distribution across chromosomes

**Usage**
```
plot_snp_distribution(marker_ranges)
```

**Arguments**
- `marker_ranges`  Genomic positions of SNPs. For a single trait, a GRanges object. For multiple traits, a GRangesList or CompressedGRangesList object, with each element of the list representing SNP positions for a particular trait. List elements must have names for proper labelling.

**Value**
A ggplot object.

**Examples**
```
data(snp_pos)
p <- plot_snp_distribution(snp_pos)
```

---

score_genes  
*Score candidate genes and select the top n genes*

**Description**
Score candidate genes and select the top n genes

**Usage**
```
score_genes(
    mined_candidates,
    hubs = NULL,
    tfs = NULL,
    pick_top = 10,
    weight_tf = 2,
    weight_hub = 2,
    weight_both = 3
)
```
simulate_windows

Arguments

mined_candidates  Data frame resulting from mine_candidates() or mine_step().
hubs  Character vector of hub genes.
tfs  Character vector of transcription factors.
pick_top  Number of top genes to select. Default: 10.
weight_tf  Numeric scalar with the weight to which correlation coefficients will be multiplied if the gene is a TF. Default: 2.
weight_hub  Numeric scalar with the weight to which correlation coefficients will be multiplied if the gene is a hub. Default: 2.
weight_both  Numeric scalar with the weight to which correlation coefficients will be multiplied if the gene is both a TF and a hub. Default: 3.

Value

Data frame with top n candidates and their scores.

Examples

data(tfs)
data(hubs)
data(mined_candidates)
set.seed(1)
scored <- score_genes(mined_candidates, hubs$Gene, tfs$Gene_ID)

simulate_windows

Simulate number of genes for each sliding window

Description

This function counts genes that are contained in sliding windows related to each SNP.

Usage

simulate_windows(
    gene_ranges,
    marker_ranges,
    windows = seq(0.1, 2, by = 0.1),
    expand_intervals = TRUE
)
Arguments

gene_ranges A GRanges object with genomic coordinates of all genes in the genome.

marker_ranges Genomic positions of SNPs. For a single trait, a GRanges object. For multiple
traits, a GRangesList or CompressedGRangesList object, with each element of
the list representing SNP positions for a particular trait.

windows Sliding windows (in Mb) upstream and downstream relative to each SNP. De-
default: seq(0.1, 2, by = 0.1).

expand_intervals Logical indicating whether or not to expand markers that are represented by
intervals. This is particularly useful if users want to use a custom interval defined
by linkage disequilibrium, for example. Default: TRUE.

Details

By default, the function creates 20 sliding windows by expanding upstream and downstream bound-
aries for each SNP from 0.1 Mb (100 kb) to 2 Mb.

Value

A ggplot object summarizing the results of the simulations.

See Also

findOverlaps-methods

Examples

data(snp_pos)
data(gene_ranges)
simulate_windows(gene_ranges, snp_pos)

snp_pos

Capsicum annuum SNPs associated with resistance to Phytophthora
root rot.

Description

The SNPs in this data set were retrieved from Siddique et al., 2019, and they are associated to
resistance to Phytophthora root rot.

Usage

data(snp_pos)

Format

A GRanges object.
References


Examples

```r
data(snp_pos)
```

<table>
<thead>
<tr>
<th>tfs</th>
<th>Pepper transcription factors</th>
</tr>
</thead>
</table>

Description

Pepper transcription factors and their families retrieved from PlantTFDB 4.0.

Usage

```r
data(tfs)
```

Format

A data frame with gene IDs in the first column and TF families in the second column.

References


Examples

```r
data(tfs)
```
Index

* datasets
  chr_length, 2
  gcn, 3
  gene_ranges, 3
  guides, 4
  hubs, 4
  mine2, 5
  mined_candidates, 5
  pepper_se, 10
  snp_pos, 14
  tfs, 15

chr_length, 2

gcn, 3

gene_ranges, 3

guides, 4

hubs, 4

mine2, 5

mine_candidates, 6

mine_step1, 7

mine_step2, 8

mine_step3, 9

mined_candidates, 5

pepper_se, 10

plot_snp_circos, 11

plot_snp_distribution, 12

score_genes, 12

simulate_windows, 13

snp_pos, 14

tfs, 15