

Package: DANDELION (via r-universe)

June 10, 2026

Type Package

Title Discovery of Candidate Disease Genes Using Trans-Regulatory Effects

Version 0.1.0

Description Implements the DANDELION method for prioritizing candidate disease-related proximal genes by integrating trans-regulatory association p-values and gene-level trait association p-values. The statistical testing framework builds on the divide-aggregate composite-null test described by Liu et al. (2022) <[doi:10.1080/01621459.2021.1914634](https://doi.org/10.1080/01621459.2021.1914634)>. The biological application and SNP/gene-based prioritization workflow are motivated by Salamone et al. (under review), ``Leveraging trans-gene regulation prioritizes central genes and pathways in asthma''. The package provides functions for identifying distal-proximal gene pairs, organizing significant pairs into genomic loci, and visualizing resulting gene networks.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports igraph

Suggests qvalue, knitr, rmarkdown

VignetteBuilder knitr

Additional_repositories <https://bioconductor.org/packages/release/bioc>

Config/pak/sysreqs libgmp-dev libxml2-dev

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

Date/Publication 2026-05-07 15:03:22 UTC

RemoteUrl <https://github.com/mxxptian/dandelion>

RemoteRef HEAD

RemoteSha b681ba94e6de30641a3d2b98bfc3d9bdb0775f5b

Contents

calc_pair.gene	2
calc_pair.snp	3
gen_fig	4
med_gene	4
print.dandelion_pairs	6
print.dandelion_result	6

Index	8
--------------	----------

calc_pair.gene	<i>Organize DANDELION Gene-Gene Results</i>
----------------	---

Description

Cleans significant DANDELION pairs when the exposure side contains genes and merges nearby exposure genes into genomic loci.

Usage

```
calc_pair.gene(
  mat.sig,
  mat.p,
  p.wes,
  gene1,
  ref.table.keep,
  eta.wgs = 1e-05,
  verbose = FALSE
)
```

Arguments

mat.sig	Matrix encoding significant pairs, returned by med_gene().
mat.p	Matrix of DANDELION p-values, returned by med_gene().
p.wes	Named numeric vector of gene-level trait association p-values.
gene1	Candidate exposure genes returned by med_gene().
ref.table.keep	Gene annotation data frame after filtering. Required columns are gene_name, Chromosome, start, and end.
eta.wgs	Significance threshold for WES genes. Default is 1e-5.
verbose	Logical. If TRUE, progress messages are printed. Default is FALSE.

Value

A list containing pairs_dact, gene_pair, sig_gene2, and non_sig_gene2.

 calc_pair.snp

 Organize DANDELION SNP-Gene Results

Description

Cleans significant DANDELION pairs when the exposure side contains SNPs. SNPs are mapped to cis genes using `uniq_snp` first. If a SNP is not present in `uniq_snp`, the function optionally maps it to the nearest or overlapping gene using `SNP.ref` and `ref.table.keep`.

Usage

```
calc_pair.snp(
  mat.sig,
  mat.p,
  p.wes,
  gene1,
  uniq_snp,
  ref.table.keep,
  eta.wgs = 1e-05,
  SNP.ref = NULL,
  verbose = FALSE
)
```

Arguments

<code>mat.sig</code>	Matrix encoding significant pairs, returned by <code>med_gene()</code> .
<code>mat.p</code>	Matrix of DANDELION p-values, returned by <code>med_gene()</code> .
<code>p.wes</code>	Named numeric vector of gene-level trait association p-values.
<code>gene1</code>	Candidate SNPs returned by <code>med_gene()</code> .
<code>uniq_snp</code>	Data frame containing known SNP-to-cis-gene mapping. Required columns are <code>SNP</code> and <code>GeneSymbol</code> .
<code>ref.table.keep</code>	Gene annotation data frame after filtering. Required columns are <code>gene_name</code> , <code>Chromosome</code> , <code>start</code> , and <code>end</code> .
<code>eta.wgs</code>	Significance threshold for WES genes. Default is 1e-5.
<code>SNP.ref</code>	Optional SNP annotation data frame with columns <code>SNP</code> , <code>SNPPos</code> , and <code>SNPChr</code> . Used only for SNPs not mapped by <code>uniq_snp</code> .
<code>verbose</code>	Logical. If TRUE, progress messages are printed. Default is FALSE.

Value

A list containing `pairs_dact`, `gene.pair`, `sig_gene2`, and `non_sig_gene2`.

gen_fig *Generate DANDELION Network Figure*

Description

Generates a network plot from DANDELION gene-pair results.

Usage

```
gen_fig(gene.pair, p.wes, eta.wgs = 1e-05, pic_dir)
```

Arguments

gene.pair	Data frame of identified gene pairs, returned by <code>calc_pair.gene()</code> or <code>calc_pair.snp()</code> . Must contain columns <code>region</code> and <code>gene2</code> .
p.wes	Named numeric vector of gene-level trait association p-values.
eta.wgs	Significance threshold for WES genes. Default is 1e-5.
pic_dir	Directory to save the figure.

Value

Invisibly returns the path to the saved PDF file.

med_gene *Apply DANDELION to Identify Candidate Disease Proximal Genes*

Description

Applies the DANDELION procedure to integrate trans-regulatory association p-values and gene-level trait association p-values. The exposure side can be either distal genes or SNPs. For clarity, exposures are referred to as `gene1`, and candidate disease proximal genes are referred to as `gene2`.

Usage

```
med_gene(
  p.trans,
  p.wes,
  ref.table,
  gene1.list,
  target.fdr = 0.1,
  dist = 5e+06,
  gene1.type = c("SNP", "Gene"),
  SNP.ref = NULL,
  n.cores = 1,
  verbose = FALSE
)
```

Arguments

<code>p.trans</code>	A numeric matrix of trans-regulatory association p-values. Rows are candidate disease proximal genes (<code>gene2</code>), and columns are exposures (<code>gene1</code>), either distal genes or SNPs. Row names and column names must be provided. (e.g., from trans-eQTL studies like eQTLGen)
<code>p.wes</code>	A named numeric vector of gene-level trait association p-values (e.g., from WES burden tests or GWAS-based gene-level tests). Note: Ensure that names (Gene Symbols) match the row names of <code>p.trans</code>
<code>ref.table</code>	A data frame containing gene annotation and genomic positions. Required columns are <code>gene_name</code> , <code>type</code> , <code>Chromosome</code> , <code>start</code> , and <code>end</code> . Chromosome should use the format <code>chr1</code> , <code>chr2</code> , etc.
<code>gene1.list</code>	A character vector of candidate exposures to analyze. These values must be a subset of <code>colnames(p.trans)</code> .
<code>target.fdr</code>	False Discovery Rate threshold. DANDELION uses this to decide which gene pairs are statistically significant. Default is 0.1.
<code>dist</code>	The cis-window size (in base pairs). Genes within this distance from the exposure will be excluded to focus on true distal (trans) effects. Default is 5e6.
<code>gene1.type</code>	Exposure type. Must be either "SNP" or "Gene".
<code>SNP.ref</code>	Optional SNP annotation data frame used when <code>gene1.type = "SNP"</code> . Required columns are <code>SNP</code> , <code>SNPPos</code> , and <code>SNPChr</code> . <code>SNPChr</code> can be 1 or <code>chr1</code> ; both formats are accepted.
<code>n.cores</code>	Number of cores for parallel execution. On non-Windows systems, values larger than 1 use <code>parallel::mclapply()</code> . On Windows, the function falls back to single-core execution.
<code>verbose</code>	Logical. If TRUE, progress messages are printed. Default is FALSE.

Value

A list with three elements: `gene1`, the analyzed exposures with at least one valid DANDELION result; `mat.sig`, a matrix encoding significant gene1-gene2 pairs; and `mat.p`, a matrix of DANDELION p-values.

Examples

```
set.seed(1)
p.trans <- matrix(runif(60, 0.001, 0.9), nrow = 12, ncol = 5)
rownames(p.trans) <- paste0("G", 1:12)
colnames(p.trans) <- paste0("E", 1:5)
p.wes <- runif(12, 0.001, 0.9)
names(p.wes) <- paste0("G", 1:12)
ref.table <- data.frame(
  gene_name = c(paste0("G", 1:12), paste0("E", 1:5)),
  type = "protein_coding",
  Chromosome = "chr1",
  start = seq_len(17) * 1e7,
  end = seq_len(17) * 1e7 + 1000
)
```

```

res <- med_gene(
  p.trans = p.trans,
  p.wes = p.wes,
  ref.table = ref.table,
  gene1.list = colnames(p.trans),
  gene1.type = "Gene"
)

```

`print.dandelion_pairs` *Print DANDELION pair results*

Description

Print DANDELION pair results

Usage

```

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

```

Arguments

`x` A `dandelion_pairs` object returned by `calc_pair.gene()` or `calc_pair.snp()`.
`...` Additional arguments, currently unused.

Value

Invisibly returns `x`.

`print.dandelion_result`
Print a DANDELION result object

Description

Print a DANDELION result object

Usage

```

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

```

Arguments

`x` A `dandelion_result` object returned by `med_gene()`.
`...` Additional arguments, currently unused.

print.dandelion_result

7

Value

Invisibly returns x.

Index

`calc_pair.gene`, 2

`calc_pair.snp`, 3

`gen_fig`, 4

`med_gene`, 4

`print.dandelion_pairs`, 6

`print.dandelion_result`, 6