Package 'GOaGO'

June 19, 2026

Title Gene Ontology enrichment analysis of gene pairs

Version 1.0.1

Description GO-a-GO annotates Gene Ontology terms that are enriched in a given set of gene pairs. The enrichment is calculated from a permutation test for overrepresentation of gene pairs that are associated with a shared term. Such gene pairs are counted for the original set of gene pairs and compared against randomized sets in which the structure of the pairs is preserved, but the gene identities (including the associated terms) are permuted.

License Artistic-2.0

Encoding UTF-8

LazyData false

Imports AnnotationDbi, BiocGenerics, BiocParallel, clusterProfiler, data.table, DOSE, GenomeInfoDb, GenomicRanges, ggplot2, ggridges, Matrix, qvalue, S4Vectors

Depends R (>= 4.4.0), methods

biocViews GO, GeneSetEnrichment

URL https://github.com/ajank/GOaGO

BugReports https://github.com/ajank/GOaGO/issues

RoxygenNote 7.3.3

Suggests BiocStyle, GenomicInteractions, ggrepel, knitr, org.Hs.eg.db, rmarkdown, rtracklayer, testthat, TxDb.Hsapiens.UCSC.hg19.knownGene

Config/testthat/edition 3

VignetteBuilder knitr

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

git_branch RELEASE_3_23

git_last_commit 04a1ba9

git_last_commit_date 2026-05-06

Repository Bioconductor 3.23

Date/Publication 2026-06-18

Author Aleksander Jankowski [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2212-6224>)

Maintainer Aleksander Jankowski <aleksander.jankowski@uw.edu.pl>

Contents

GOaGO-package *GOaGO: Gene Ontology enrichment analysis of gene pairs*

Description

GO-a-GO annotates Gene Ontology terms that are enriched in a given set of gene pairs. The enrichment is calculated from a permutation test for overrepresentation of gene pairs that are associated with a shared term. Such gene pairs are counted for the original set of gene pairs and compared against randomized sets in which the structure of the pairs is preserved, but the gene identities (including the associated terms) are permuted.

Author(s)

Maintainer: Aleksander Jankowski <aleksander.jankowski@uw.edu.pl> (ORCID)

See Also

Useful links:

- https://github.com/ajank/GOaGO
- Report bugs at https://github.com/ajank/GOaGO/issues

annotateAnchors *Associate interaction anchors to the nearest TSSes*

Description

Interaction anchors are associated to TSSes as follows. Each anchor is associated to all the TSSes it overlaps. If there is no such overlap, then the anchor is associated to all TSSes with the shortest distance to the anchor, if this distance is not larger than `maxDistanceToTSS`.

Usage

```
annotateAnchors(
  anchors,
  transcripts = NULL,
  tss = NULL,
  keyType = NULL,
  maxDistanceToTSS = -1
)
```

Arguments

`anchors`	object of class `GRanges`
`transcripts`	`TxDb` annotation or other `GRanges` object
`tss`	object of class `GRanges` as returned by `convertTranscriptsToTSS`
`keyType`	type of gene identifiers, such as "ENTREZID" or "ENSEMBL", if it cannot be determined from metadata of `transcripts` or `tss`
`maxDistanceToTSS`	maximal distance to extend the search for nearest TSS outside the anchor, or -1 (the default) to skip the extension

Details

Either `transcripts` or `tss` must be provided, but not both.

Value

A data table with columns `interactionID` (index of the anchor), `chrom`, `start`, `end` (coordinates of the anchor), `geneID` (gene identifier from ‘transcripts‘ or ‘tss‘), `tss` (TSS position) and `strand` (TSS strand).

See Also

`convertTranscriptsToTSS`

Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)

# take only the transcripts of coding genes by ensuring that the coding
# sequence strand is not NA
transcripts <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene,
    columns = "gene_id", filter = list(cds_strand = c("-", "+"))
)

tss <- convertTranscriptsToTSS(transcripts)
gr <- GRanges("chr1", IRanges(c(42001, 890001), c(62000, 900000)))

# note that anchors are associated to TSSes outside the anchor only if there
# are no TSSes overlapping the anchor
annotateAnchors(gr, tss, maxDistanceToTSS = 10e3)

# this may yield more associations to TSSes outside the anchors
annotateAnchors(gr + 10e3, tss)
```

---

annotateInteractions *Associate interactions to gene pairs*

Description

Both interaction anchors are associated to TSSes as described in annotateAnchors. Briefly, each anchor is associated to all the TSSes it overlaps, or to all closest TSSes up to maxDistanceToTSS if there is no such overlap. The annotation of an interaction is a Cartesian product of annotations for both anchors.

Usage

```
annotateInteractions(
  interactions,
  transcripts = NULL,
  tss = NULL,
  keyType = NULL,
  maxDistanceToTSS = -1
)
```

Arguments

interactions	object of class Pairs (of GRanges) or GenomicInteractions
transcripts	TxDb annotation or other GRanges object
tss	object of class GRanges as returned by convertTranscriptsToTSS
keyType	type of gene identifiers, such as "ENTREZID" or "ENSEMBL", if it cannot be determined from metadata of transcripts or tss
maxDistanceToTSS	maximal distance to extend the search for nearest TSS outside the anchor, or -1 (the default) to skip the extension

Details

Either `transcripts` or `tss` must be provided, but not both.

Value

A data table with columns `interactionID` (index of the interaction), `chrom1`, `start1`, `end1`, `chrom2`, `start2`, `end2` (coordinates of both anchors), `geneID1`, `geneID2` (gene identifiers from 'transcripts' or 'tss' for both anchors), `tss1`, `tss2` (TSS position for both anchors), `strand1` and `strand2` (TSS strand for both anchors).

See Also

`convertTranscriptsToTSS`, `annotateAnchors`

Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)

# take only the transcripts of coding genes by ensuring that the coding
# sequence strand is not NA
transcripts <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene,
    columns = "gene_id", filter = list(cds_strand = c("-", "+"))
)

fpath <- system.file("extdata", "GM12878_loops.bedpe.gz", package = "GOaGO")
pairs <- rtracklayer::import(fpath, genome = "hg19")
annotateInteractions(pairs, transcripts, maxDistanceToTSS = 10e3)
```

`as.data.frame` *Coerce a* `GOaGO-result` *object to a data frame*

Description

Coerce a `GOaGO-result` object to a data frame

Usage

```
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```

Arguments

`x`	The object to coerce.
`row.names, optional, ...`	Not used, inherited from `base::as.data.frame()`.

Value

A data frame of the enriched Gene Ontology terms, with the following columns: `ONTOLOGY`, `ID`, `Description` (all of the GO term), `Count` (number of input gene pairs sharing the given term), `PairRatio` (fraction of input gene pairs sharing the given term), `BgRatio` (fraction of permuted gene pairs sharing the given term), `FoldEnrichment` (quotient of the two fractions), `pvalue`, `p.adjust`, `qvalue`.

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
as.data.frame(goago)
```

---

convertTranscriptsToTSS

Convert gene transcripts to Transcription Start Sites

Description

Convert a `GRanges` object with gene transcripts to a `GRanges` object with gene TSSes of length 1 bp, with duplicate rows removed. Assumes that gene identifiers are provided in one of the metadata columns, either as a vector (possibly containing `NA` values) or a `CharacterList`. The function is idempotent, i.e. can be applied multiple times without changing the result.

Usage

```
convertTranscriptsToTSS(
  transcripts,
  geneid_column = c("gene_id", "GENEID", "geneID")
)
```

Arguments

`transcripts`	`TxDb` annotation or other `GRanges` object
`geneid_column`	A character vector of recognized names for the metadata column in `transcripts` that contains gene identifiers. If none or more than one is found, an error is raised.

Value

A `GRanges` object with gene identifiers in metadata column `geneID` being a vector.

Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)

# take only the transcripts of coding genes by ensuring that the coding
# sequence strand is not NA
transcripts <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene,
    columns = "gene_id", filter = list(cds_strand = c("-", "+"))
)

convertTranscriptsToTSS(transcripts)
```

`DotPlot` *Dotplot of the enriched Gene Ontology terms*

Description

By default, plots the most enriched terms, with fold enrichment on the X-axis, point size indicating the number of gene pairs sharing the given term, and point color – the adjusted p-value.

Usage

```
DotPlot(
  object,
  minCount = 5,
  x = "FoldEnrichment",
  color = "p.adjust",
  size = "Count",
  showCategory = 10,
  orderBy = "FoldEnrichment",
  decreasing = TRUE,
  font.size = 12,
  label_format = 50
)
```

Arguments

`object`	GO-a-GO results of class `GOaGO-result`
`minCount`	plot only the GO terms that are associated to at least the given number of gene pairs
`x`	Variable for X-axis, one of `"FoldEnrichment"`, `"PairRatio"` and `"Count"`.
`color`	Variable used to color enriched terms, e.g. `"pvalue"`, `"p.adjust"` or `"qvalue"`.
`size`	Variable used to scale the sizes of points, one of `"FoldEnrichment"`, `"PairRatio"` and `"Count"`.
`showCategory`	number of terms to display or a vector of terms
`orderBy`	The order of the Y-axis, one of `"FoldEnrichment"`, `"PairRatio"` and `"Count"`.

decreasing	logical. Should the orderBy order be increasing or decreasing?
font.size	font size
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels. By default wraps names longer than 50 characters.

Value

A ggplot object that can be further customized using the ggplot2 package.

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
DotPlot(goago)
```

---

genePairsGM12878 *Gene pairs associated with chromatin loops in GM12878 cell line*

Description

The dataset is based on 9,448 chromatin loops identified in human cell line GM12878 as peaks in Hi-C contact maps. Of these chromatin loops, 1,581 overlapped at least one gene Transcription Start Site (TSS) at both loop anchors. As some loop anchors overlapped multiple TSSes, possibly of different genes, the dataset contains all combinations for these loops, yielding a total of 2,339 gene pairs, of which 1,743 pairs are unique and do not contain the same gene twice.

Usage

```
genePairsGM12878
```

Format

A data frame with 2,339 rows and 13 columns:

interactionID loop identifier

chrom1 chromosome of loop anchor 1

start1 start coordinate of loop anchor 1

end1 end coordinate of loop anchor 1

geneID1 Entrez identifier of the gene associated to loop anchor 1

tss1 TSS coordinate of the associated gene

strand1 strand ("+" or "-") of the associated gene

chrom2 chromosome of loop anchor 2

start2 start coordinate of loop anchor 2

end2 end coordinate of loop anchor 2

geneID2 Entrez identifier of the gene associated to loop anchor 2

tss2 TSS coordinate of the associated gene

strand2 strand ("+" or "-") of the associated gene.

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

References

Rao, S. S., Huntley, M. H., *et al.* (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7), 1665-80.

`GOaGO` *Gene Ontology enrichment analysis in a set of gene pairs*

Description

Given a data frame of gene pairs, this function will return the enriched Gene Ontology terms after FDR control.

Usage

```
GOaGO(
  genePairs,
  OrgDb,
  keyType = NULL,
  ont = "MF",
  minCount = 1,
  numPermutations = 10000,
  universe,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
  qvalueCutoff = 0.2,
  minGSSize = 10,
  maxGSSize = 500
)
```

Arguments

`genePairs`	a data frame with columns `geneID1` and `geneID2` containing gene identifiers; column `pairID` will also be used if provided.
`OrgDb`	OrgDb
`keyType`	type of gene identifiers, such as "ENTREZID" or "ENSEMBL", if it cannot be determined from metadata of `genePairs`

ont	one of "BP", "MF", and "CC" subontologies, or "ALL" for all three
minCount	cutoff for number of pairs that share a GO term for this term to be considered
numPermutations	number of permutations performed in the enrichment test
universe	a set of background genes. If missing, all the genes from all the gene pairs will be used as background.
pvalueCutoff	adjusted p-value cutoff on enrichment tests to report
pAdjustMethod	one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"
qvalueCutoff	q-value cutoff on enrichment tests to report as significant. Tests must pass (i) pvalueCutoff on unadjusted p-values, (ii) pvalueCutoff on adjusted p-values, and (iii) qvalueCutoff on q-values to be reported.
minGSSize	minimal size of genes annotated for testing
maxGSSize	maximal size of genes annotated for testing

Value

A GOaGO-result object.

See Also

GOaGO-result-class

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
show(goago)
```

GOaGO-accessors *Accessors and show method for* GOaGO-result *objects*

Description

Accessors and show method for GOaGO-result objects

Usage

```
genePairs(object)

keyType(object)

organism(object)

show(object)
```

Arguments

object of class GOaGO-result

Value

genePairs returns a data frame with the input gene pairs, with the columns geneID1, geneID2 and pairID.

keyType returns the type of gene identifiers, such as "ENTREZID" or "ENSEMBL".

organism returns the scientific name (i.e. genus and species, or genus and species and subspecies) of the organism.

show displays the object, and returns an invisible NULL.

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
show(goago)

genePairs(goago)
keyType(goago)
organism(goago)
```

GOaGO-result-class *An S4 class to represent the results of GO-a-GO enrichment analysis*

Description

An S4 class to represent the results of GO-a-GO enrichment analysis

Slots

result A data frame of the enriched Gene Ontology terms, with the following columns: ONTOLOGY, ID, Description (all of the GO term), Count (number of input gene pairs sharing the given term), PairRatio (fraction of input gene pairs sharing the given term), BgRatio (fraction of permuted gene pairs sharing the given term), FoldEnrichment (quotient of the two fractions), pvalue, p.adjust, qvalue.

pvalueCutoff adjusted p-value cutoff on enrichment tests

pAdjustMethod p-value adjustment method

qvalueCutoff q-value cutoff on enrichment tests

minCount cutoff for number of pairs that share a GO term for this term to be considered

numPermutations number of permutations performed in the enrichment test

minGSSize minimal size of genes annotated for testing

`maxGSSize` maximal size of genes annotated for testing

`organism` scientific name of the organism

`ontology` one of "BP", "MF", and "CC" subontologies, or "ALL" for all three

`keyType` type of gene identifiers, such as "ENTREZID" or "ENSEMBL"

`genePairs` A data frame with the input gene pairs, with the columns `geneID1`, `geneID2` and `pairID`.

`pairTerms` A data frame linking the enriched Gene Ontology terms with the input gene pairs, with the columns `pairID` and `ID` (of the GO term).

`permutedResult` A data frame with the columns `ID` (of the GO term) and `Count`, keeping the numbers of permuted gene pairs sharing the term as obtained in every random permutation.

`universe` a set of background genes

`RidgePlot` *Ridgeplot of the sampling distributions for the randomized gene pairs*

Description

Ridgeplot of the sampling distributions of numbers of gene pairs sharing each enriched Gene Ontology term, obtained for the randomized gene pairs.

Usage

```
RidgePlot(
  object,
  minCount = 5,
  showCategory = 10,
  orderBy = "FoldEnrichment",
  decreasing = TRUE,
  font.size = 12,
  label_format = 50
)
```

Arguments

`object`	GO-a-GO results of class `GOaGO-result`
`minCount`	plot only the GO terms that are associated to at least the given number of gene pairs
`showCategory`	number of terms to display or a vector of terms
`orderBy`	The order of the Y-axis, one of `"FoldEnrichment"`, `"PairRatio"` and `"Count"`.
`decreasing`	logical. Should the `orderBy` order be increasing or decreasing?
`font.size`	font size
`label_format`	a numeric value sets wrap length, alternatively a custom function to format axis labels. By default wraps names longer than 50 characters.

Value

A ggplot object that can be further customized using the ggplot2 package.

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
RidgePlot(goago)
```

---

termGenePairs *Extract the enriched Gene Ontology terms along with gene pairs sharing them*

Description

Extract the enriched Gene Ontology terms along with gene pairs sharing them

Usage

```
termGenePairs(object, OrgDb = NULL)
```

Arguments

object	of class GOaGO-result
OrgDb	OrgDb to map gene identifiers to gene symbols

Value

A data frame similar to returned by as.data.frame(object), but including all the gene pairs sharing each enriched Gene Ontology term, one gene pair in each row. Additional columns include pairID, geneID1, geneID2 and any other columns provided in genePairs argument to GOaGO. If OrgDb is provided, geneSymbol1 and geneSymbol2 will also be added.

See Also

as.data.frame,GOaGO-result-method

Examples

```
library(org.Hs.eg.db)
data("genePairsGM12878")

goago <- GOaGO(genePairsGM12878, keyType = "ENTREZID", OrgDb = org.Hs.eg.db)
termGenePairs(goago, OrgDb = org.Hs.eg.db)
```

`uniqueGenePairs` *Extract unique gene pairs from the data frame provided*

Description

Given a data frame of gene pairs, this function will return the unique pairs of genes, removing loops (gene pairs containing the same gene twice) and duplicates. Note that gene pair (A, B) is a duplicate of (B, A).

Usage

```
uniqueGenePairs(genePairs)
```

Arguments

`genePairs`	a data frame with columns `geneID1` and `geneID2` containing gene identifiers; column `pairID` will also be used if provided.

Value

A data frame with columns `pairID`, `geneID1` and `geneID2`. If loops or duplicates were removed, a warning will alert you. If column `pairID` was not provided in `genePairs`, an integer vector equal to `seq_len(nrow(result))` will be used.

Index