Package 'CompGO'

October 8, 2015

Title An R pipeline for .bed file annotation, comparing GO term
enrichment between gene sets and data visualisation

Description This package contains functions to accomplish several tasks. It is
able to download full genome databases from UCSC, import .bed files easily,
annotate these .bed file regions with genes (plus distance) from
aforementioned database dumps, interface with DAVID to create functional
annotation and gene ontology enrichment charts based on gene lists (such as
those generated from input .bed files) and finally visualise and compare
these enrichments using either directed acyclic graphs or scatterplots.

Version 1.4.0

License GPL-2

Depends RDAVIDWebService

Imports rtracklayer, Rgraphviz, ggplot2, GenomicFeatures,
TxDb.Mmusculus.UCSC.mm9.knownGene, pcaMethods, reshape2,
pathview

biocViews GeneSetEnrichment, MultipleComparison, GO, Visualization

NeedsCompilation no

Author Sam D. Bassett [aut],
Ashley J. Waardenberg [aut, cre]

Maintainer Ashley J. Waardenberg <A.Waardenberg@victorchang.edu.au>

R topics documented:

---

annotateBedFromDb   *Annotate .bed file to genes*

Description

Wrapper for transcriptsByOverlaps(). Returns a GRanges with the gene and transcript ids associated with the input .bed regions. Sometimes it is necessary to expand the search window a bit, because not all .bed regions directly overlap with a transcription start site, so the 'window' parameter is provided to accomplish this.

Usage

```
annotateBedFromDb(pathToBed = NULL, gRanges = NULL, db = NULL,
  window = 5000)
```

Arguments

pathToBed	The system path to a .bed file (directory + file name)
gRanges	If the user has a .bed file already loaded in R, they can supply it here as a GRanges object rather than re-importing it
db	A TxDb object containing the transcripts of the organism (required)
window	The window around a .bed region to search for genes, default 5kb

Value

A GRanges object with corresponding EntrezGene IDs in gene_id column, plus transcript IDs in tx_id

Examples

```
library(TxDb.Mmusculus.UCSC.mm9.knownGene)
txdb = TxDb.Mmusculus.UCSC.mm9.knownGene
data(bed.sample)
range = GRanges(seqnames=bed.sample$chr, IRanges(start=bed.sample$start, end=bed.sample$end))
x = annotateBedFromDb(gRanges = range, db = txdb)
x
```

---

bed.sample	*A sample of 25 rows from a .bed file of mm9 regions*

Description

25 regions from a .bed file for use in example code, contains regions from mm9

Usage

```
bed.sample
```

Format

A data.frame with 25 obs. of 3 variables: chromosome, start position, end position

---

compareZscores	*Compare the Z scores of individual GO terms between two input annotation charts*

Description

Accepts two fnAnot charts as args, does z score and p value calculations on them and returns a data.frame with important data. A flag, geneInfo, is provided in case the user wants to get information about the intersection and union of genes corresponding to the individual GO terms. Importantly, this function does some implicit thresholding: only terms with a minimum of 'cutoff' genes are compared, and any term present in one list but not the other is discarded.

Usage

```
compareZscores(setA, setB, geneInfo = FALSE, cutoff = 10)
```

Arguments

setA	FunctionalAnnotationChart to compare
setB	FunctionalAnnotationChart to compare
geneInfo	Whether to add gene intersection and union info to the data.frame
cutoff	The minimum number of genes to threshold terms by

Value

A data.frame with columns: Term, Zscore.A, Zscore.B, ComparedZ, Pvalue (optionally geneUnion, geneIntersect as well, which are comma-separated strings).

Examples

```
data(funChart1)
data(funChart2)
cz = compareZscores(funChart1, funChart2)
str(cz)
cz = compareZscores(funChart1, funChart2, geneInfo = TRUE)
str(cz)
```

---

doZtrans.single *Z transform a single functional annotation chart from DAVID*

Description

Decomposes each GO term in a functional annotation chart (returned from getFnAnot_genome()) to its Z-score. These tables can be merged for clustering

Usage

```
doZtrans.single(x, name)
```

Arguments

x	The functional annotation chart to apply the transformation to
name	(optional) The name to give the Z-score column; if not supplied, name is derived from the input variable

Value

A data.frame of GO terms and Z-scores

Examples

```
# Load example fnAnot charts from DAVID:
data(funChart1)
zscore = doZtrans.single(funChart1)
str(zscore)
```

---

gata4	*A .bed file containing genomic ranges sampled from gata4*

Description

1000 genomic ranges sampled from gata4 binding sites identified in HL-1 cells (mm9)

Usage

```
gata4
```

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

---

getFnAnot_genome	*Get the functional annotation chart of a gene list using DAVID*

Description

Uploads a gene list to DAVID, then performs a GO enrichment analysis. Requires registration with DAVID first here. Returns a DAVIDFunctionalAnnotationChart object which can be easily coerced into a data.frame. DAVID does some automatic thresholding on results. For Z-score standardisation, we found it useful to get DAVID to return all possible annotations despite non-significant P-values and perform our own thresholding.

Usage

```
getFnAnot_genome(geneList, david = NULL, email = NULL,
  idType = "ENTREZ_GENE_ID", listName = "auto_list", count = 1L,
  PVal = 1, background = NULL, bgIdType = NULL, bgListName = NULL,
  getKEGG = FALSE)
```

Arguments

geneList	Either a list of genes or a GRanges result from annotateBedFromDb to upload and functionally enrich
david	An RDAVIDWebService object can be passed to the function so a new one doesn't have to be requested each time

email	If david==NULL, an email must be supplied. DAVID requires (free) registration before users may interact with their WebService API. This can be accomplished online (here), then the registered email supplied here.
idType	The type of gene IDs being uploaded (MGI, Entrez,...)
listName	The name to give the list when it's uploaded to the WebService
count	Minimum number of genes per GO term
PVal	P-value threshold for GO terms
background	If you want to perform enrichment against a specific background instead DAVID's default (whole genome), supply it here
bgIdType	If the background gene ID type is different from the gene list, enter it here
bgListName	If you want to give the background a name, enter it here
getKEGG	TRUE if you want to download KEGG pathway information as well as GO

Value

Returns a DAVIDFunctionalAnnotationChart after generating it by comparing the supplied gene list to the full genome as a background

Examples

```
## not run because registration is required
## visit http://david.abcc.ncifcrf.gov/webservice/register.htm to register
## Not run:
## You can either supply the registered email:
fnAnot = getFnAnot_genome(exp1$gene_id,
   email = "your.registered@email.com",
   idType="ENTREZ_GENE_ID", listName="My_gene_list-1")
## Or create a DAVIDWebService object with the email:
david = DAVIDWebService$new(email = "your.registered@email.com")
fnAnot = getFnAnot_genome(entrezList, david = david)

## End(Not run)
```

---

mef2a   *A .bed file containing genomic ranges sampled from mef2a*

Description

1000 genomic ranges sampled from mef2a binding sites identified in HL-1 cells (mm9)

Usage

```
mef2a
```

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

---

nkx25   *A .bed file containing genomic ranges sampled from nkx25*

Description

1000 genomic ranges sampled from NKX2-5 binding sites identified in HL-1 cells (mm9)

Usage

    nkx25

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

---

p300   *A .bed file containing genomic ranges sampled from p300*

Description

1000 genomic ranges sampled from p300 binding sites identified in HL-1 cells (mm9)

Usage

    p300

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

PCAplot   *Plot PCA given an input list of fnAnot charts*

Description

Given a list of functional annotation charts, this function outputs a PCA plot

Usage

```
PCAplot(input)
```

Arguments

input   A list of functional annotation charts.

---

plotDendrogam   *Interactive plotting function for groups of GO terms*

Description

Given a list of functional annotation charts and optionally an output directory, this function can output dendrograms, PCA analysis plots and a correlation matrix to make large-scale comparisons easy.

Usage

```
plotInteractive(input, outDir = NULL, prefix = NULL, pdf = TRUE)
```

Arguments

input   A list of functional annotation charts.

outDir   The directory to save plots to.

prefix   The prefix to append to each file, if any.

pdf   If true, plots will be pdfs. If false, pngs.

| plotDendrogram | *Plot dendrogram given an input list of fnAnot charts* |

Description

Given a list of functional annotation charts, this function outputs a dendrogram

Usage

```
plotDendrogram(input)
```

Arguments

input	A list of functional annotation charts.

| plotPairwise | *Generates a scatterplot of two sets of GO terms based on DAVID P-values* |

Description

Generates a -log10 scatterplot of two sets of GO terms by p-value or corrected p-value with linear fit and correlation. Also includes a Jaccard metric for gene overlap within each GO term. Useful as an overall metric of gene list similarity. NOTE: The plotZScores function is more statistically sound, you should use that instead of this.

Usage

```
plotPairwise(setA, setB, cutoff = NULL, useRawPvals = FALSE,
  plotNA = TRUE, model = "lm", ontology = NULL)
```

Arguments

setA	DAVIDFunctionalAnnotationChart object to compare
setB	DAVIDFunctionalAnnotationChart object to compare
cutoff	The p-value or adjusted p-value to use as a cutoff
useRawPvals	If false, uses adjusted p-values, otherwise uses the raw ones
plotNA	If true, any GO term present in only one list is considered to have a p-value of 1 in the other; otherwise, it is simply removed
model	The model to use when plotting linear fit, default 'lm'
ontology	If a specific ontology (MF, BP, CC) is wanted rather than all terms, supply it here as a string

Examples

```
data(funChart1)
data(funChart2)
plotPairwise(funChart1, funChart2)
```

---

plotTwoGODags   *Plots a directed acyclic graph of GO terms from two different sources*

Description

Plots a directed acyclic graph of GO terms from two different sources, using colour to show intersection and difference. This is useful to see the specific functional differences between gene lists, complementing the overall metric of gene list similarity

Usage

```
plotTwoGODags(setA, setB, ont = "BP", cutoff = 0.01, maxLabel = NULL,
  fullNames = TRUE, Pvalues = TRUE)
```

Arguments

setA	A DAVIDFunctionalAnnotationChart object
setB	A DAVIDFunctionalAnnotationChart object
ont	The ontology to use, one of BP, MF and CC
maxLabel	Maximum length of GO term to print
cutoff	The PValue cutoff to use
fullNames	Whether to print the full GO term label or just the GO id
Pvalues	Whether to print P-values alongside each label

References

Fresno, C. and Fernandes, E. (2013) RDAVIDWebService: An R Package for retrieving data from DAVID into R objects using Web Services API. <http://david.abcc.ncifcrf.gov/>

Examples

```
data(funChart1)
data(funChart2)
plotTwoGODags(funChart1, funChart2)
```

---

plotZRankedDAG	*Plot a directed acyclic graph (DAG) based on the corrected Pvalues generated from comparing two sets of Z scores.*

Description

This function accepts two functional annotation charts as input, performs a comparison on them using compareZscores() and plots a DAG based on the results. The saturation of each node is computed based on the Pvalue, such that the more significant values are darker in colour.

Usage

```
plotZRankedDAG(setA, setB, ont = "BP", n = 100, maxLabel = NULL,
  fullNames = TRUE, Pvalues = TRUE)
```

Arguments

setA	FunctionalAnnotationChart to compare
setB	FunctionalAnnotationChart to compare
ont	The gene ontology category for which to calculate enrichment
n	The number of top-ranked Pvalues to compare
maxLabel	The maximum number of characters in a node's label
fullNames	Whether to print the full GO term label or just the GO id
Pvalues	Whether to print P-values alongside each label

Examples

```
## Not run:
data(funChart1)
data(funChart2)
plotZRankedDAG(funChart1, funChart2, n = 50)

## End(Not run)
```

---

plotZScores	*Performs z transform on two sets of GO terms and plots scatterplot of result*

Description

Generates a scatterplot of z transformed GO terms and plots the result along with the Jaccard metric for each GO term and linear fit + correlation.

Usage

```
plotZScores(setA, setB, cutoff = NULL, plotAbs = TRUE, plotNA = FALSE, model = "lm")
```

Arguments

setA	DAVIDFunctionalAnnotationChart object to compare
setB	DAVIDFunctionalAnnotationChart object to compare
plotAbs	Whether to plot the absolute values of z-scores or the raw values
plotNA	Whether to remove NAs entirely or set all NAs to 0
model	The model to use when plotting linear fit, default 'lm'
cutoff	If you want to apply a Benjamini corrected P-value cutoff to each list before generating Z scores, supply it here

Examples

```
data(funChart1)
data(funChart2)
plotZScores(funChart1, funChart2)
```

---

slidingJaccard	*Plot two functional annotation charts using a sliding Jaccard coefficient*

Description

This function compares two functional annotation charts using a sliding Jaccard coefficient - a ranked list of P-values is produced, and a sliding window is used to find the Jaccard coefficient of two charts at different cutoffs of the top n terms. This is useful to determine where the majority of overlapping terms is located, and can also be used to compare Jaccard profiles between multiple (up to 4) sets if C and D are supplied.

Usage

```
slidingJaccard(setA, setB, increment = 50, setC = NULL, setD = NULL)
```

Arguments

setA	A DAVIDFunctionalAnnotationChart to compare
setB	A DAVIDFunctionalAnnotationChart to compare
increment	The number of terms (n) to increment for each sliding window
setC	A DAVIDFunctionalAnnotationChart to compare, optional
setD	A DAVIDFunctionalAnnotationChart to compare, optional

Examples

```
data(funChart1)
data(funChart2)
slidingJaccard(funChart1, funChart2, 50, FALSE)
```

---

srf	*A .bed file containing genomic ranges sampled from srf*

Description

1000 genomic ranges sampled from srf binding sites identified in HL-1 cells (mm9)

Usage

```
srf
```

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

---

tbx5	*A .bed file containing genomic ranges sampled from tbx5*

Description

1000 genomic ranges sampled from tbx5 binding sites identified in HL-1 cells (mm9)

Usage

```
tbx5
```

Format

A data frame with 1000 observations of 3 variables: chromosome, start position, end position.

Source

He, A. (2011) *Co-occupancy by multiple cardiac transcription factors identifies transcriptional enhances active in heart*. PNAS 108(14), 5632-5637

---

viewKegg   *Compare KEGG pathways between two functional annotation charts*

Description

viewKegg uses pathview to compare the gene lists visually by KEGG pathway. You can either supply a pathway id or the function will pick the most differentially enriched pathway between the two inputs. As functional annotation charts don't have differential gene expression information, a boolean scale is used - genes in the pathway are coloured green if from setA, yellow if from both, and red if from setB. We recommend you supply a working directory, as pathview will download an XML and PNG file as well as output an additional PNG of the pathway.

Usage

```
viewKegg(setA, setB, keggTerm = NULL, species = NULL, workingDir = NULL,
  sortByCount = FALSE, ...)
```

Arguments

setA	FunctionalAnnotationChart to compare
setB	FunctionalAnnotationChart to compare
keggTerm	If a specific KEGG pathway is of interest, input the name here; otherwise, the most differentially expressed pathway will be used.
species	The program can usually figure out the species from the KEGG terms, but if it can't, supply the species ID here. From pathview vignette, run 'data(bods); bods' to find species codes.
workingDir	The directory to output into. Recommended, since pathview will put a few different files there each time.
sortByCount	Set TRUE if you want the function to automatically choose the pathway with the most number of genes
...	Arguments to be passed to pathview

Value

Output from pathview: a list of 2, plot.data.gene and plot.data.cpd

Examples

```
## Not run:
# Since this function requires writing to a directory, it won't be run here
data(funChart1)
data(funChart2)
viewKegg(funChart1, funChart2)

## End(Not run)
```

| zTransformDirectory | *Z-score transformation of DAVID functional annotation charts in a supplied directory* |

Description

Given a directory of functional annotation charts, this function iterates over them and generates Odds Ratio, St. Error and Z scores. This is useful for batch processing, as all the charts can be written to disk somewhere then iterated over by this function automatically. Two options are provided for dealing with absent terms: either the NAs are set as 0 (a pseudo-representation of a Z-score with no enrichment), or incomplete rows are removed. The final table can be used for clustering analyses.

Usage

```
zTransformDirectory(inputDir, cutoff = NULL, pattern = NULL,
  removeNA = FALSE)
```

Arguments

inputDir	The directory to search for functional annotation charts
pattern	The regex pattern to match files in inputDir
cutoff	Reduce the computation to the top n GO terms ranked by variance
removeNA	True to only generate the Z-transform table based on GO terms common to all input enrichment analyses, False to set all NAs as 0

Value

Returns a data.frame of z scores, ORs and SEs

Examples

```
## Not run:
#not run as dir required
z.merge = zTransformDirectory("./fnAnot_charts", pattern = "-fnAnot.txt")
# To plot a dendrogram based on Z-scores:
d <- cor(abs(z.merge[2:(ncol(z.merge)-1)]))
dist.cor <- hclust(dist(1-d), method="complete")
plot(dist.cor, xlab="Complete linkage", sub = NA)

## End(Not run)
```

Index