Package 'GSCA'

January 19, 2021

Type Package

Title GSCA: Gene Set Context Analysis

Version 2.20.0

Date 2015-12-8

Author Zhicheng Ji, Hongkai Ji

Maintainer Zhicheng Ji <zji4@jhu.edu>

Description GSCA takes as input several lists of activated and repressed genes. GSCA then searches through a compendium of publicly available gene expression profiles for biological contexts that are enriched with a specified pattern of gene expression. GSCA provides both traditional R functions and interactive, user-friendly user interface.

License GPL(>=2)

LazyLoad yes

Imports graphics

Depends shiny, sp, gplots, ggplot2, reshape2, RColorBrewer, rhdf5, R(>= 2.10.0)

Suggests Affyhgu133aExpr, Affymoe4302Expr, Affyhgu133A2Expr, Affyhgu133Plus2Expr

biocViews GeneExpression, Visualization, GUI

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

git_branch RELEASE_3_12

git_last_commit 34a9327

git_last_commit_date 2020-10-27

Date/Publication 2021-01-18

R topics documented:

GSCA-package *GSCA: Gene Set Context Analysis*

Description

GSCA analyzes biological contexts enriched within given patterns of geneset expression activity. GSCA takes as input several lists of activated and repressed genes. Though the input genesets could contain any gene which interest users, they are usually dervied from ChIP-chip or ChIP-seq (ChIPx) and gene expression data in one or more biological systems, for example TF target genes (genes that are both TF-bound in the ChIPx data and differentially expressed in the gene expression data). Then GSCA uses the given genesets to scan through a compendium of gene expression profiles constructed from publicly available gene expression data to search for patterns of geneset expression activity specified by the users. The final output of GSCA is a ranked table of biological contexts that are significantly enriched with the specified pattern of geneset expression activity. After the initial GSCA analysis, users can further study the predicted biological contexts and related contexts in more detail using the tabSearch function to search for contexts of interest in the human or mouse compendium, and the GSCAeda function to visualize and test for differences in geneset expression activities of the recovered contexts. Further functions to help annotate peaks and construct TF target genes are also provided if users are interested in exploring enriched biological contexts in given TF expression and target gene activity. Besides traditional R functions, GSCA also provides a user-friendly interactive user interface based on R shiny. Users can run GSCAui function to run the UI in the web browser on their own computer (need to install shiny and GSCAdata package) or go to http://spark.rstudio.com/jzc19900805/GSCA/ to run the UI on shiny server (only a web browser is required, do not need to install GSCA, GSCAdata or R).

Details

Package:	GSCA
Type:	Package
Version:	2.1.0
Date:	2015-12-8
License:	GPL-2

Author(s)

Author: Zhicheng Ji, Hongkai Ji Maintainer: Zhicheng Ji <zji4@jhu.edu>

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

annotatePeaks *Annotate ChIPx peaks with genes by Entrez GeneIDs*

Description

This function finds all genes that overlap with each peak detected from TF ChIP-chip or ChIP-seq data. Assigned genes are assumed to be genes bound by the TF.

Usage

```
annotatePeaks(inputfile, genome, up = NULL, down = NULL)
```

Arguments

inputfile	A data.frame where each row corresponds to a peak. The first column is the chromosome on which the peak is found (e.g., chr1) and the second and third columns are the peak starting and ending sites.
genome	Should be one of 'hg19', 'hg18', 'mm9', or 'mm8' genome. More genomes may be supported in future versions of GSCA.
up	Region upstream of the TSS. A gene will be annotated to a peak if the region upstream to downstream of each gene TSS, as defined by the up and down arguments, overlap with the peak.
down	Region downstream of the TSS. A gene will be annotated to a peak if the region upstream to downstream of each gene TSS, as defined by the up and down arguments, overlap with the peak.

Details

A gene will be annotated to a peak if the region upstream to downstream of each gene TSS, as defined by the up and down arguments, overlap with the peak.

Value

Returns a data.frame with the same columns as the input data.frame and an additional column containing the Enterz GeneIDs for all genes that overlap with the peak. Multiple genes will be separated with ';' and '-9' will be reported if no genes are found.

Author(s)

Zhicheng Ji, Hongkai Ji

References

Chen X, Xu H, Yuan P, Fang F et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. Cell 2008 Jun 13;133(6):1106-17.

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

Examples

```
### Read in example ChIP-seq analyzed data output from GSE11431
### for Oct4 in ESCs directly downloaded from NCBI GEO
path <- system.file("extdata",package="GSCA")
inputfile <- read.delim(paste(path,"GSM288346_ES_Oct4.txt",sep="/"), header=FALSE,stringsAsFactors=FALSE)

### Note that 1st column is chr, 2nd and 3rd columns are starting and ending sites of peaks
### Remaining columns are other output from the peak detection algorithm
head(inputfile)

### annotatePeaks only requires the first 3 columns
annon.out <- annotatePeaks(inputfile,"mm8",10000,5000)
head(anon.out)
```

`ConstructTG` *Construct target genes for a TF using TF-bound genes and differentially expressed genes from ChIP-chip or ChIP-seq and TF perturbation gene expression data.*

Description

This function requires users to first analyze their own ChIP-chip and ChIP-seq data to detect significant peaks and then annotate the peaks with their corresponding regulated target genes using the annotatePeaks function in the GSCA package. Users must also use the limma package to detect differentially expressed genes in their gene expression data (preprocessing and noramzliation can be done with any algorithm the user desires), then the resulting output needs to be annotated into Entrez GeneIDs. Finally, with both inputs ConstructTG will identify the activated and repressed TF target genes.

Usage

```
ConstructTG(annonPeaksOut, limmaOut)
```

Arguments

`annonPeaksOut`	Output from the annotatePeaks function in the GSCA package. Contains the genes that correspond to the significant peaks detected from TF ChIP-chip or ChIP-seq data.
`limmaOut`	Differential expression output from the limma package, and requires the first column of the data.frame to contain the EntrezGeneIDs that match the microarray probeset IDs.

Details

This function is designed as one method to allow users to construct target genes after obtaining a list of significant peaks from ChIP-chip or ChIP-seq data and differential expression results from using limma to anaylze their microarray data. It is not designed to be flexible to account for all methods to obtain TF-bound and/or differentially expressed genes. Users can choose to manually intersect their own TF-bound and differentially expressed genes by classifying activated genes as genes, whose expression increases when the TF expression increases and repressed genes as genes, who expression decreases when the TF expression increases. Note, that significant cutoffs for peaks and differentially expressed genes need to be already applied prior to input.

Value

Returns a list with two items:

PosTG	Activated TF target genes
NegTG	Repressed TF target genes

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

Examples

```
### Read in example ChIP-seq analyzed data output from GSE11431
### for Oct4 in ESCs directly downloaded from NCBI GEO
path <- system.file("extdata",package="GSCA")
chipxfile <- read.delim(paste(path,"GSM288346_ES_Oct4.txt",sep="/"),
                        header=FALSE,stringsAsFactors=FALSE)

### annotate each peak with the corresponding gene target
annon.out <- annotatePeaks(chipxfile,"mm8",10000,5000)

### Read in example limma output from gene expression data obtained
### by analyzing Oct4 RNAi knockdown gene with RMA then limma
### from the raw CEL files in GSE4189
### The first column contains the Entrez GeneID for each probeset ID
### annotated using the mouse4302.db package in Bioconductor.
gp.out <- read.delim(paste(path,"Pou5f1_E14TG2a_GSE4189_Limma.txt",sep="/"),
                             stringsAsFactors=FALSE)

ConstructTG(annon.out,gp.out)
```

---

geneIDdata *Homologene data*

Description

Homologene data to support conversion of ENTREZ gene ID and gene name between human and mouse species.

References

http://www.ncbi.nlm.nih.gov/homologene

Examples

```
data(geneIDdata)
```

GSCA *GSCA*

Description

The function takes as input several lists of activated and repressed genes. It then searches through a compendium of publicly available gene expression profiles for biological contexts that are enriched with a specified pattern of gene expression.

Usage

```
GSCA(genedata,pattern,chipdata,scaledata=F,Pval.co=0.05,directory=NULL)
```

Arguments

genedata	A data.frame with three columns specifying the input genesets. Each row specifies an activated or repressed gene in a geneset. First column: character value of geneset name specified by the user, could be any name easy to remember e.g. GS1,GS2,...; Second column: numeric value of Entrez GeneID of the gene; Third column: numeric value of single gene weight when calculating the activity level of the whole geneset. Positive values for activated gene and negative values for repressed gene. Here, activated gene means that increases in expression of the gene also increases the overall activity of the whole geneset, while increases in expression of the repressed genes will decrease the overall activity of the whole geneset.
pattern	A data.frame with four columns indicating the activity patterns corresponding to the given genedata. Each row specifies activity pattern for one geneset. First column: character value of the same geneset name used in genedata, each geneset name in genedata should appear exactly once in this column. Second column: character value of whether high or low activity of the whole geneset is interested. "High" stands for high activity and "Low" stands for low activity. Third column: character value of which cutoff type is going to be used. 3 cutoff types can be specified: "Norm", "Quantile", or "Exprs". If cutoff type is "Norm", then the fourth column should be specified as p-value between 0 and 1, where the geneset expression cutoff will correspond to the specified p-value (one-sided) based on a fitted normal distribution; If cutoff type is "Quantile", then the fourth column should be specified as a desired quantile between 0 and 1, where the geneset expression cutoff will correspond to the specified quantile. Finally, if cutoff type is "Exprs", the geneset expression cutoff will be equal to the value given in the fourth column. Fourth column: numeric value of cutoff value based on different cutoff types specified in the third column.
chipdata	A character value of 'hgu133a', 'hgu133A2', 'hgu133Plus2' or 'moe4302'. This argument specifies which compendium to use. Requires the corresponding data package.
scaledata	logical value indicating whether expression data for each gene should be scaled across samples to have mean 0 and variance 1.
Pval.co	A numeric value specifying the adjusted p-value cutoff. Only the biological contexts with significant enrichment above the adjusted p-value cutoff will be reported in the final ranked table output.

`directory` Either null or a character value giving a directory path. If directory is not null, then additional follow-up GSCA analyses will be performed and stored in the folder specified by directory. If directory is null then no additional follow-up GSCA analyses will be performed.

Details

GSCA requires as input user-specified genesets together with their corresponding activity patterns. Each geneset contained the Entrez GeneID of activied and repressed genes. Activated gene means that increases in expression of the gene also increases the overall activity of the whole geneset, while repressed gene means that increases in expression of the gene decreases the overall activity of the whole geneset.

GSCA also requires activity patterns of the genesets. Users can choose either high or low level of activity for each geneset. Cutoffs are given by the users to determine what activity level should be considered high or low. There are three types of cutoffs available: normal, quantile and expression value. For normal cutoff type, a specified p-value (one-sided) based on a fitted normal distribution will be used as cutoff, and all samples having p-value larger(smaller) than this p-value will be considered having high(low) expression activity in a certain geneset.Likewise, for quantile cutoff type, a quantile will be used as cutoff. As for cutoff type of expression level, a numeric value will be directly used as cutoff.

GSCA then searches through the compendium for all samples that exhibit the specified activity pattern of interest. For example, if activity patterns of all genesets are set to be high, then GSCA will find all samples in the compendium that have greater geneset expression levels than the respective cutoffs. Since each of the samples correspond to different biological contexts, the Fisher's exact test will then be used to test the association between each biological context and the geneset activity pattern of interest based on the number of samples in each biological context that exhibits the specified geneset activity pattern of interest.

The final output is a ranked table of biological contexts enriched with the geneset activity pattern of interest. The p-values are also adjusted by the Bonferroni correction.

If directory is not null, then GSCA will peform detail analyses for all contexts in each of the experimental IDs in the final GSCA results table. For each of the experiment IDs, tabSearch will be run to locate all contexts in the compendium for that experiment ID, and then GSCAeda will be run using the same genedata and pattern as input specific to the contexts recovered by tabSearch. See GSCAeda for more details. Note, this automated process could be time-consuming and produce a lot of files and directories.

Value

Returns a list with

`Ranking` Data.frame of ranked table of biological contexts significantly enriched with the specified geneset activity pattern. It includes information of Ranking, number of samples exhibiting the given activity pattern, total number of samples, fold change values, adjusted p-values, name of biological context and corresponding experiment ID.

`Score` Numeric matrix of geneset expression values for each sample in the compendium. Each row stands for a certain geneset and each column stands for a certain sample.

`Pattern` Data.frame of geneset activity pattern. The same as the input value.

`Cutoff` Numeric vector of cutoff values calculated for each geneset based on the input pattern.

SelectedSample	Numeric vector of all samples that exhibits the given geneset activity pattern.
Totalgene	Numeric vector of total number of genes use to calculate the geneset activity in each geneset
Missinggene	Numeric vector of number of genes that do not have corresponding expression measuremnts on the platform
Species	Character value of the species analyzed.

If directory is not null, then pdf and csv files containing the GSCAeda follow-up analysis results and plots in the directory folder will also be returned.

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

Examples

```
## First load the TF target genes derived from Oct4 ChIPx data
## in embryonic stem cells. The data is in the form of a list
## where the first item contains the activated (+) target genes in
## Entrez GeneID format and the second item contains the repressed (-)
## target genes in Entrez GeneID format.
data(Oct4ESC_TG)

## We want to analyze Oct4, so we need to specify the EntrezGeneID for Oct4
## and input the activated (+) and repressed (-) target genes of Oct4.
## Constucting the input genedata required by GSCA. There are two genesets
## one is the TF and another is the TF target genes. Note that constructing genedata
## with many genesets could be laborious, so using the interactive UI is recommended to
## easily start up the analysis.
activenum <- length(Oct4ESC_TG[[1]])
repressnum <- length(Oct4ESC_TG[[2]])
Octgenedata <- data.frame(gsname=c("GS1",rep("GS2",activenum+repressnum)),gene=c(18999,Oct4ESC_TG[[1]],Oct4

## We are interested in the pattern that TF and its target genes are all highly expressed.
## We also need to define how high the cutoffs should be such
## that each cutoff corresponds to the p-value of 0.1
## based on fitted normal distributions.
## Constructing pattern required by GSCA, all geneset names in genedata should appear
## exactly once in the first column
Octpattern <- data.frame(gsname=c("GS1","GS2"),acttype="High",cotype="Norm",cutoff=0.1,stringsAsFactors=FAL

## Lastly, we specify the chipdata to be "moe4302" and the significance of enriched
## biological contexts must be at least 0.05 to be reported.
Octoutput <- GSCA(Octgenedata,Octpattern,"moe4302",Pval.co=0.05)

## The first item in the list 'Octoutput[[1]]' contains the ranked table, which
## can then be saved. Additionally, we may be interested in plotting the results
## to visualize the enriched biological contexts within given geneset activity.
## Here, N specifies the top 5 significant biological contexts.
## Since plotfile is NULL, the plot directly shows up in R.
```

```
## Check GSCAplot for more details.
GSCAplot(Octoutput,N=5,plotfile=NULL,Title="GSCA plot of Oct4 in ESC")

## If you would like detailed follow-up analyses to be automatically performed
## for the Oct4 analyses in ESCs, just specify a file directory.
## Check GSCAeda for more details.

Octoutput <- GSCA(Octgenedata,Octpattern,"moe4302",Pval.co=0.05,directory=tempdir())

## All output will be stored in the specified directory.
## This process may be time-consuming and generate a lot of files.
## Alternatively, see GSCAeda for more info on manual alternatives.
```

GSCAeda *GSCA follow-up exploratory data analysis*

Description

GSCAeda is used to further study GSCA significant predictions in more detail to obtain additional insight into biological function. GSCAeda requires users to first run the tabSearch function to identify the biological contexts of interest. By default, GSCAeda will run automatically after an initial GSCA analysis by searching for all contexts related to the experimentID for each significant GSCA prediction. Alternatively, users can use GSCAeda by itself to further study any geneset or biological contexts of interest that are found in the compendium. The output of GSCAeda are multiple plots displaying the geneset activity values and genes of interest in the input biological contexts. Also included are the usual GSCA analysis results table showing the enrichment of each contexts for the geneset activity pattern of interest, t-test results (t-statistics and p-values) for all pair-wise combinations of inputted contexts in each geneset, and a summary of raw geneset activity values for each context of interest. Users can then use the raw geneset activity values for further statistical analyses if desired.

Usage

```
GSCAeda(genedata,pattern,chipdata,SearchOutput,scaledata=F,Pval.co=0.05,Ordering="Average",Title
```

Arguments

genedata	A data.frame with three columns specifying the input genesets. Each row specifies an activated or repressed gene in a geneset. First column: character value of geneset name specified by the user, could be any name easy to remember e.g. GS1,GS2,...; Second column: numeric value of Entrez GeneID of the gene; Third column: numeric value of single gene weight when calculating the activity level of the whole geneset. Positive values for activated gene and negative values for repressed gene. Here, activated gene means that increases in expression of the gene also increases the overall activity of the whole geneset, while increases in expression of the repressed genes will decrease the overall activity of the whole geneset.
pattern	A data.frame with four columns indicating the activity patterns corresponding to the given genedata. Each row specifies activity pattern for one geneset. First column: character value of the same geneset name used in genedata, each geneset name in genedata should appear exactly once in this column. Second column:

character value of whether high or low activity of the whole geneset is interested. "High" stands for high activity and "Low" stands for low activity. Third column: character value of which cutoff type is going to be used. 3 cutoff types can be specified: "Norm", "Quantile", or "Exprs". If cutoff type is "Norm", then the fourth column should be specified as p-value between 0 and 1, where the geneset expression cutoff will correspond to the specified p-value (one-sided) based on a fitted normal distribution; If cutoff type is "Quantile", then the fourth column should be specified as a desired quantile between 0 and 1, where the geneset expression cutoff will correspond to the specified quantile. Finally, if cutoff type is "Exprs", the geneset expression cutoff will be equal to the value given in the fourth column. Fourth column: numeric value of cutoff value based on different cutoff types specified in the third column.

`chipdata` A character value of 'hgu133a', 'hgu133A2', 'hgu133Plus2' or 'moe4302'. This argument specifies which compendium to use. Requires the corresponding data package.

`scaledata` logical value indicating whether expression data for each gene should be scaled across samples to have mean 0 and variance 1.

`SearchOutput` Output of the tabSearch function. More specifically, a data frame where the 1st column is the ExperimentIDs (GSE ids), the 2nd column is the SampleTypes, and the 3rd column is the sample count for each SampleType.

`Pval.co` A numeric value specifying the adjusted p-value cutoff. Only the biological contexts with significant enrichment above the adjusted p-value cutoff will be reported in the final ranked table output.

`Ordering` A character value of either one geneset name or 'Average'. If Ordering is one geneset name, the plot of geneset activity values and heatmap of the t-statistics/pvalues will be ordered from the highest to lowest according the Ordering geneset activity value. If Ordering is 'Average', the plots and heatmap will be organized by the average rank across all geneset activity values.

`Title` Title of the plot, will appear on the top of the plot.

`outputdir` Either null or a character value giving the directory in which GSCAeda will save the output files.

Details

GSCAeda is designed to be used in combination with tabSearch after an initial GSCA analysis. GSCAeda is used to further study each predicted biological context in more detail by comparing the functional activity across related contexts through the geneset activities. To do so, GSCAeda requires users to specific genedata, pattern, species, pval cutoff, and the search results from tabSearch containing the list of biological contexts of interest. Then GSCAeda will calculate the mean and standard deviation of each geneset activity value for each inputted context, and will perform t-tests comparing the mean geneset activity values for all pair-wise combinations of inputted contexts, and test for enrichment of the geneset activity pattern of interest. The results will be shown in several plots and tables(number of files varying with number of given genesets), along with the raw geneset activity values for further followup statistical analyses. Check the value part of this help file to see how GSCAeda saves the outputs. For information on the GSCA parameters, see the GSCA help file which explains in more detail how functional enrichment of a geneset activity pattern of interest is tested.

Value

If outputdir is specified, GSCAeda will first produce a boxplot depicting the distribution of all geneset activities in different biological contexts of interest. Then, for each geneset GSCAeda will

produce tow heatmaps showing respectively the t-statistics and p-values obtained from the t-tests testing the mean of geneset activity for each pair-wise combination of the input biological contexts. Finally, GSCAeda will output two csv files. The first one contains the raw geneset activity values for each input context and the second one contains the mean and standard deviation of the geneset activity values for each context, the GSCA enrichment test results, and the p-values/t-statistics of the t-tests. If outputdir is NULL, all plots and the result table will be directly displayed in the R console.

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

See Also

GSCA

Examples

```
library(GSCA)

## Load example STAT1 target genes defined ChIP-seq and literature
data(STAT1_TG)

## Construct genedata and pattern using the same way as GSCA
Statgenenum <- length(STAT1_TG)
Statgenedata <- data.frame(gsname=c("GS1",rep("GS2",Statgenenum)),gene=c(6772,STAT1_TG),weight=1,stringsAsF
Statpattern <- data.frame(gsname=c("GS1","GS2"),acttype="High",cotype="Norm",cutoff=0.1,stringsAsFactors=FA

## Find all contexts in human compendium from GSE7123
GSE7123out <- tabSearch("GSE7123","hgu133a")

## Run GSCAeda
GSCAeda(Statgenedata,Statpattern,"hgu133a",GSE7123out,Pval.co=0.05,Ordering="Average",Title=NULL,outputdir=

## To save the results, instead of displaying in R console, specifiy an outputdir argument

GSCAeda(Statgenedata,Statpattern,"hgu133a",GSE7123out,Pval.co=0.05,Ordering="Average",Title=NULL,outputdir=
```

GSCAplot *Visualize GSCA output*

Description

GSCAplot visualizes the output from GSCA. For one geneset, GSCAplot makes histograms of geneset activities for all samples and samples in each of most significantly enriched biological contexts. For two genesets, GSCAplot makes a scatter plot of sample activities of first geneset versus sample activities of second geneset plotting, and the most significantly enriched biological contexts are highlighted in the plot. For more than two genesets, GSCAplot produces two heatmap

according to geneset activities. The first heatmap shows the geneset activities of all samples and indicates which samples belong to enriched biological contexts. The second heatmap shows the geneset activities of samples exhibiting given geneset activity pattern, and the most significantly enriched biological contexts are highlighted.

Usage

```
GSCAplot(GSCAoutput,N=5,plotfile=NULL,Title=NULL)
```

Arguments

GSCAoutput	Exact output from GSCA.
N	N is a numeric value ranging from 1 to 5. It specifies the number of top-ranked biological contexts to plot from the GSCA analysis.
plotfile	A character value specifying the path to save the GSCA plot. If plotfile is null, the plot will not be saved and will appear directly in R console.
Title	A character value specifying the title of the plot.

Details

GSCAplot is a plotting function that acts as an easy-to-use tool to visualize the GSCA output. For one geneset, GSCAplot uses histogram() to first plot a histogram of geneset activities for all samples in the compendium, then plot N histograms of geneset activities for samples in each of top N most significantly enriched biological contexts. For two genesets, GSCAplots uses plot() to make a scatterplot of all samples in the compendium where x-axis is the activity of the first geneset and y-axis is the activity of the second geneset. Then it highlights the top N most significantly enriched biological contexts in different colors and types. Cutoff of the two genesets will also be represented on the scatterplot as one vertical and one horizontal dotted line. For more than two genesets, GSCAplots uses heatmap.2() from gplots package to plot two heatmaps. In the first heatmap, geneset activities of all samples in the compendium will be shown. A color legend will be drawn on the left upper corner of the heatmap so that users will know the corresponding activity value each color represents. Above the heatmap there is a color bar of light and dark blue indicating which samples exhibit the specific geneset activity pattern. In the second heatmap, geneset activity of all samples which exhibit the specific geneset activity pattern will be shown. A color bar above the heatmap uses different colors to indicate top N most significantly enriched biological contexts. A color legend will also appear in the left upper corner of the heatmap. If plotfile is not null, then instead of showing the plots directly in the R console, GSCAplot will save the plots to the designated filepath as a pdf file. Note that because there are a lot of samples in both human and mouse compendiums, drawing the first heatmap (and sometimes the second heatmap) could take a lot of time especially a large number of genesets are given. GSCAplot only supports a predefined geneset activity pattern and basic plotting options. Users are encouraged to use the interactive UI if they want to interactively determine the geneset activity pattern, gain more powerful plotting options and further customize their plots.

Value

A plot consisting of several histograms, a scatterplot or two heatmaps will be returned, depending on numbers of genesets users give.

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

Examples

```
## Constructing genedata and pattern.
## Example of mouse gene Gli1,Gli2 and Gli3, all members of GLI-Kruppel family. Their corresponding Entrez GeneI
gligenedata <- data.frame(gsname=c("Gli1","Gli2","Gli3"),gene=c(14632,14633,14634),weight=1,stringsAsFactor
glipattern <- data.frame(gsname=c("Gli1","Gli2","Gli3"),acttype="High",cotype="Norm",cutoff=0.1,stringsAsFa

## Case of one geneset: a set of histograms
## Note that for N too large sometimes there is figure margins too large error.
## Decrease N or try to enlarge the plotting area in R console.
oneout <- GSCA(gligenedata[1,],glipattern[1,],"moe4302")
GSCAplot(oneout,N=2)

## Case of two genesets: a scatterplot
twoout <- GSCA(gligenedata[-3,],glipattern[-3,],"moe4302")
GSCAplot(twoout)

## Case of three genesets: two heatmaps, press Enter to switch to the second heatmap
## May take some time, be patient
threeout <- GSCA(gligenedata,glipattern,"moe4302")
GSCAplot(threeout)

## Same plots in designated file path, FILE, which is a pdf file.
## If you want to further customize output plots, for example changing
## range of x-axis, changing titles or altering display of enriched
## biological contexts, please check out the interactive user interface.

GSCAplot(oneout,plotfile=tempfile("plot",fileext=".pdf"),N=2,Title="Demo of one geneset plot")
GSCAplot(twoout,plotfile=tempfile("plot",fileext=".pdf"),Title="Demo of two genesets plot")
GSCAplot(threeout,plotfile=tempfile("plot",fileext=".pdf"),Title="Demo of three genesets plot")
```

`GSCAui` *Launch GSCA interactive User Interface*

Description

GSCAui initiates in the web browser an interactive user interface of GSCA built using R shiny. This user interface enables users to easily perform nearly all standard GSCA functions in GSCA package, and provides more powerful and useful options to specify geneset activity patterns, investigate interested biological contexts and customize output plots and tables. For a complete user manual of GSCAui, please refer to the user manual included in the user interface.

Usage

```
GSCAui()
```

Details

The purpose of building GSCA interactive user interface is to provide an easy way for all users to perform analysis tools offered by GSCA, even though the users do not have any prior knowledge in computer programming or statistics. GSCAui provides users handy ways to input their original dataset into the program. Users who do not have much experience using R may find themselves having difficulties building genedata and pattern datasets required by standard GSCA functions. In comparison, GSCAui offers more covenient ways to directly type in gene IDs and specify parameters like cutoff types using pull down menus. Users can also check instantly how many genes they inputted are recorded in a given compendium and decide what geneset to be used in further analysis process. GSCAui also provides users more flexible and direct means to specify geneset activity patterns. For different number of geneset, GSCAui will automatically generate control panels which are most suitable for users to interactively choose the geneset activity patterns. Users can not only specify geneset activity patterns using traditional GSCA options, but they can also choose geneset activity pattern on histograms, scatterplots and heatmaps by point and click, which makes the process easier and more explicit. In addition, GSCAui offers more powerful analysis and plotting options. Both p-value and foldchange cutoffs can be given interactively to select the enriched biological contexts. Besides displaying top ranked enriched biological contexts, users can also select specific biological contexts to be displayed on the plot. Finally, users can specify plotting details like x-axis range and titles of the plots if they want to keep the plots for future use. Thanks to the shiny server, users can type in the URL: http://spark.rstudio.com/jzc19900805/GSCA/ to directly launch the UI in their web browser. This does not require any dependent R packages or even R itself installed on users computer. All required is a web browser and the URL. Please check the user manual in the UI for more complete explanations.

Value

A user interface will be shown in users' default web browser. R console will start listenting to a random port.

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

See Also

GSCA

Examples

```
## Running this will launch the UI in users' default web browser.
## Not run:
GSCAui()

## End(Not run)
```

Oct4ESC_TG — *Oct4 activated (+) and repressed (-) target genes in embryonic stem cells*

Description

List of Oct4 target genes derived from ChIP-seq and gene expression data from embryonic stem cells (ESCs). Activated target genes are the first item in the list and repressed target genes are the second item in the list.

Usage

```
data(Oct4ESC_TG)
```

Format

The format is: List of 2 $: chr [1:519] "100678" "106298" "14609" "12468" ... $: chr [1:337] "246703" "15441" "70579" "20333" ...

Details

Oct4 target genes are defined as genes that are both predicted to be TF-bound in E14 ESCs and differentially expressed after Oct4 knockdown via RNAi in E14TG2a ESCs.

Source

Chen X, Xu H, Yuan P, Fang F et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. Cell 2008 Jun 13;133(6):1106-17.

Loh YH, Wu Q, Chew JL, Vega VB et al. The Oct4 and Nanog transcription network regulates pluripotency in mouse embryonic stem cells. Nat Genet 2006 Apr;38(4):431-40.

References

http://www.ncbi.nlm.nih.gov/geo/

Examples

```
data(Oct4ESC_TG)
```

STAT1_TG — *STAT1 activated (+) target genes defined from experimental ChIP-seq data and literature survey.*

Description

List of STAT1 target genes derived from ChIP-seq data in Hela cells and further refined by making sure each target gene was further supported by experiments in literature as described in GSE15353. No represed target genes were defined.

Usage

```
data(STAT1_TG)
```

Format

The format is: chr [1:23] "9636" "2537" "2633" "1435" "103" "3433" "3434" ...

Details

STAT1 target genes are defined as TF-bound from Hela ChIP-seq data and then further verified as target genes through literature survey. This procedure is described in GSE15353.

Source

Robertson G, Hirst M, Bainbridge M, Bilenky M et al. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. Nat Methods 2007 Aug;4(8):651-7.

References

http://www.ncbi.nlm.nih.gov/geo/

Examples

```
data(STAT1_TG)
```

---

tabSearch *Searches through GPL96, GPL1261, GPL570 or GPL571 compendium data for biological contexts of interest.*

Description

tabSearch requires users to provide keyword(s), the species, and either 'AND' or 'OR'. Then the function uses grep and the keywords to iteratively search for biological contexts or experiment IDs that match the keywords, where 'AND' requires all recovered contexts to satisfiy all keywords and 'OR' requires all recovered contexts to match at least one keyword.

Usage

```
tabSearch(keyword, chipdata, option = "OR")
```

Arguments

keyword	A character vector of biological context words or experiment IDs. e.g. 'liver' or 'GSE7123'.
chipdata	A character value of 'hgu133a', 'hgu133A2', 'hgu133Plus2' or 'moe4302'. This argument specifies which compendium to use. Requires the corresponding data package.
option	Either 'AND' or 'OR' to specify whether the recovered contexts need to be found by ALL keywords (AND) or found by at least one keyword (OR).

Details

If the users want to search for a specific list of contexts, simply input the contexts as a character vector, where each element is a different context. Alternatively, the contexts can also be a series of keywords in short-hand. The 'AND' option is primarly used when users want to search for contexts from a specific experiment. In most cases 'OR' should be used.

After tabSearch finishes running, it will return a list of contexts that match the inputted keywords and parameters. Users can then further study these contexts for activity of given gensets using the function GSCAeda.

Value

A data frame consisting of three columns. The 1st column is the experiment ID, the 2nd column is the biological context label, and the 3rd column is the number of samples for each biological context.

Author(s)

Zhicheng Ji, Hongkai Ji

References

George Wu, et al. ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. Bioinformatics 2013 Apr 23;29(9):1182-1189.

Examples

```
library(GSCA)
## Search for all contexts in GSE7123 in hgu133a
tabSearch("GSE7123","hgu133a")

## Search for all contexts labeled 'fetal' or 'liver' in moe4302
tabSearch(c("Fetal","Liver"),"moe4302")

## Search for all contexts labeled 'fetal liver' AND in GSE13044 in moe4302
tabSearch(c("Fetal","GSE13044"),"moe4302","AND")
```

Index