Package 'dmGWAS'

December 4, 2010

Type Package

Title dmGWAS package

Version 2.0

Date 2010-12-01

Author Peilin Jia

Maintainer Peilin Jia <peilin.jia@vanderbilt.edu>

Depends R (>= 2.9.0), igraph

Description dmGWAS is an R package searching for dense modules in PPI network using GWAS dataset

License GPL (>= 2)

URL http://bioinfo.mc.vanderbilt.edu/dmGWAS.html

R topics documented:

dms *Dense module searching function*

Description

dms constructs a node-weighted PPI network, performs dense module searching, generates simulation data from random networks, normalizes module scores using simulation data, removes unqualified modules, and orders resultant modules according to their significance.

Usage

```
dms(network, gene2weight, d=2, r=0.1)
```

Arguments

network	A data frame containing a symbolic edge list of the PPI network.
gene2weight	A data frame containing two columns: the first is unique gene identifier (should be coordinate with the node symbol used in pina); the second is P values from GWAS. this can also be generated from SNP2Gene.match.R
d	An integer used to define neighborhood genes to perform searching. Default is 2.
r	A float indicating the cutoff for increasement during module expanding process. Greater r will generate smaller module. Default is 0.1.

Value

A list containing all important data including the node-weighted network used for searching, resultant dense module list, module score matrix containing Zm and Zn, and randomization data for normalization. A file named "RESULT.list.RData" is also automatically saved in the working folder for future record.

References

Jia P, Zheng S, Sun J, Long J, Zheng W, and Zhao Z. (2010). Discovering combined causal signals from genome-wide association studies (GWAS) by network module based approach.

Examples

```
#res.list = dms(network, gene2weight, d=2, r=0.1)
```

dualEval *module selection based on two GWAS datasets*

Description

Given multiple GWAS datasets, dense module searching can be performed independently for each GWAS dataset, generating two "RESULT.list.RData" files seperately. dualEval will then take the two result files as input and perform dense module selection based on the two files.

Usage

```
dualEval(resfile1, resfile2)
```

Arguments

resfile1	the discovery file "RESULT.list.RData" resulted by dms.R using the discovery GWAS dataset.
resfile2	the evaluation file "RESULT.list.RData" resulted by dms.R using the evaluation GWAS dataset.

References

Jia P, Zheng S, Sun J, Long J, Zheng W, and Zhao Z. (2010). Discovering combined causal signals from genome-wide association studies (GWAS) by network module based approach.

Examples

```
resfile1 = "/path/to/RESULT.list.RData" # using the discovery GWAS dataset
resfile2 = "/path/to/RESULT.list.RData" # using the evaluation GWAS dataset
#dualEval(resfile1, resfile2)
```

moduleChoose *choosing modules using seed gene list*

Description

Given the seeds of modules to be selected, this function chooses the modules for each seed and returns the combined subnetwork of all the selected modules. Subnetwork can also be plotted if plot=TRUE.

Usage

```
moduleChoose(seed.nodes, res.list, plot = FALSE)
```

Arguments

seed.nodes	seeds of modules to be selected
res.list	the results from dms.R
plot	to plot the subnetwork or not

Value

an object of list contains: subnetwork the combined subnetwork module.list gene lists of each selected module

References

Jia P, Zheng S, Sun J, Long J, Zheng W, and Zhao Z. (2010). Discovering combined causal signals from genome-wide association studies (GWAS) by network module based approach.

Examples

```
#moduleChoose(seedgenes, res.list, plot=T)
```

PCombine *Combine multiple SNPs for one gene*

Description

Compute a summarizing P value for each gene from multiple SNPs related to the gene in a GWAS dataset.

Usage

```
PCombine(gene.map, method = "NULL")
```

Arguments

gene.map	a data.frame containing Gene-SNP-P information such as one returned by "SNP2Gene.match"
method	one of "simes", "smallest", "fisher", "gwbon"

Details

To generate a summarized P value for each gene based on multiple SNPs represented in a GWAS dataset, several methods are implemented by PCombine: simes (the Simes method), fisher (Fisher's combined method), smallest (using the smallest P value for the gene), or gwbon (using the smallest P value for the gene after gene-wise Bonferroni correction).

Value

gene2weight	an object of data.frame, including gene and its summarized P value

References

Luo L, Peng G, Zhu Y, Dong H, Amos CI, Xiong M. (2010). Genome-wide gene and pathway analysis. Eur J Hum Genet, Epub ahead of print

Examples

```
#gene2weight = PCombine(gene.map, method="smallest")
```

---

simpleChoose *simply choose the top N modules*

Description

choose the top N ranked modules based on Zn

Usage

```
simpleChoose(res.list, top = 100, plot=FALSE)
```

Arguments

`res.list`	list. the object returned by dms.R
`top`	integer. number of modules to be chosen.
`plot`	boolean. whether to draw the subnetwork or not.

Value

an object of list contains: subnetwork the combined subnetwork module.list gene lists of each selected module

Examples

```
#load('RESULT.list.RData') #RESULT.list.RData is a file generated directly by dms
#simpleChoose(res.list, top = 100)
```

SNP2Gene.match *Map SNPs to genes*

Description

map SNPs to genes

Usage

```
SNP2Gene.match(assoc.file, snp2gene.file, boundary = 20)
```

Arguments

assoc.file	a file including SNPs and their P values. Ideally it should be a file generated directly by PLINK, e.g., using –assoc option. A header line is needed. Specifically, the column for SNP ids must be named "SNP" and the column for P values mube be named "P".
snp2gene.file	a file including SNP-gene mapping infomation. Four columns are needed in this file, "SNP", "rs", "dist", "gene". Users can downloaded this file directly from our website. Or use PLINK to do the mapping using –gene-report option, and format the result file then into a 4-column file. We prepared two files for AffyMatrix 5.0 and 6.0 arrays. Users may choose the corresponding file for their SNP id.
boundary	the distance within with a SNP is considered to be related to the gene.

Value

gene.map data.frame. columdns are coming in the following order: SNP, rs, dist, gene,

Examples

```
#gene.map=SNP2Gene.match(assoc.file="gwas.assoc", snp2gene.file="GenomeWideSNP_5.na30.annot.
```

statResults *statistic description of the results*

Description

statResults provide statistic description of the network features of the resultant modules in one round. Modules will be sorted according to their score, Zn. By walking down this sorted list, one module will be added in each step, and combined to a subnetwork. Clustering coefficient, average shortest path, and average degree will be ploted for each step.

Usage

```
statResults(res.list, top=500)
```

Arguments

res.list	a list of modules resulted during the process of dense module searching. It's also saved in the file 'RESULT.list.RData' generated by dms.R
top	the number of top ranked modules to be checked.

References

Jia P, Zheng S, Sun J, Long J, Zheng W, and Zhao Z. (2010). Discovering combined causal signals from genome-wide association studies (GWAS) by network module based approach.

Examples

```
#load('RESULT.list.RData')
#statResults(res.list, top=500)
```

---

zn.permutation *Analysis of permutation data to compute nominal P value for each module*

Description

This function will compute Zn for each module in the permutation data and count the number of permutations that have higher Zn than the real case for each module. A nominal P is computed then to indicate the association of each module with the disease of interest.

Usage

```
zn.permutation(module.list, gene2snp, gene2snp.method='smallest', original.file, pe
```

Arguments

module.list	a list of modules to test using permutation data.
gene2snp	a dataframe describing the gene ~ snp relationship coming in two columnds, gene and snp.
gene2snp.method	one of "simes", "smallest", "fisher", "gwbon". must be the same as used in PCombine.
original.file	path to the assoc file of the real data.
permutation.dir	path to the folder where permutation data files (generated by PLINK) are located.

Format

The format is: chr "zn.permutation.R"

Details

This function will compute Zn for each module in the permutation data and count the number of permutations that have higher Zn than the real case for each module. A nominal P is computed to indicate the association of each module with the disease of interest.

Value

zn.matrix: matrix, each row for one module and each column for one permutation data. Zn(ij) in the matrix indicates Zn of ith module in jth permutation (j=1:1000). The last column is the Zn for real case of modules.

References

Jia P, Zheng S, Sun J, Long J, Zheng W, and Zhao Z. (2010). Discovering combined causal signals from genome-wide association studies (GWAS) by network module based approach.

Examples

```
#zn.permutation(module.list, gene2snp, perm.split.dir)
```

Index