GraphAT

March 24, 2012

`Phenoclusters` *Yeast Gene-Knockout Fitness Data Cluster Memberships*

Description

This data set contains cluster memberships for yeast genes clustered using fitness deficiency scores from gene knockout experiments from Giaever et al. Nature (2002). The 3000 most variable genes were clustered using k-means with 30 clusters

Usage

```
data(Phenoclusters)
```

Format

A matrix whose rows are the 3000 genes and whose two columns are gene name and cluster membership number.

Source

http://gobi.lbl.gov/YeastFitnessData

References

Giaever, G. et al. 2002 "Functional profiling of the Saccharomyces cerevisiae genome." Nature **418**, 387–391.

Examples

```
data(Phenoclusters)

## Compute the adjacency matrix for the corresponding cluster graph:
phenoMat<-clust2Mat(Phenoclusters[,2])
```

causton *Yeast mRNA Expression Data*

Description

This data set contains mRNA expression from a microarray experiment involving yeast grown under a variety of altered environments (e.g. acid, heat, sorbitol, etc.)

Usage

```
data(causton)
```

Format

A matrix whose rows are the 6015 genes and whose columns are the 45 experimental conditions.

Source

http://web.wi.mit.edu/young/environment

References

Causton, H. C. et al. 2001 "Remodeling of Yeast Genome Expression in Response to Environmental Changes." Molecular Biology of the Cell **12**, 323–337.

Examples

```
data(causton)

## Find the 3000 most variable genes, according to sd/mean:

varMeas<-function(vec) sd(vec)/mean(vec)
variability<-apply(causton,1,varMeas)

rks<-rank(variability)

causton3000<-causton[rks>length(rownames(causton))-3000,]
```

cellcycle *Cell-Cycle Cluster Matrix*

Description

An adjacency matrix in which

Usage

```
data(ccCM)
```

Format

`ccCM` is a symmetric matrix with 2885 columns and 2885 rows.

`nNamescc` is a vector of 2885 gene names.

Details

Cho, et al. discuss the k means clustering of 2885 Saccharomyces genes into 30 clusters with measurements taken over two synchronized cell cycles. `nNamescc` is a vector of the 2885 gene names. `ccCM` is an adjacency matrix in which a "1" in the ith row and jth column indicates that gene i and gene j belong to the same cluster. All other entries are 0. These data are integrated with phenotypic data and GO data in Balasubramanian, et al (2004).

Source

Balasubramanian R, LaFramboise T, Scholtens D, Gentleman R. (2004) A graph theoretic approach to integromics - integrating disparate sources of functional genomics data

References

Cho, et al. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell, 2, 65-73.

Tavazoie, et al. (1999) Systematic determination of genetic network architecture. Nature Genetics, 22, 281-285.

Examples

```
data(ccCM)
```

`clust2Mat` *Function to compute adjacency matrix of cluster graph given a vector of cluster memberships*

Description

Given a list of cluster memberships, this function computes the adjacency matrix for the associated cluster graph. The adjacency matrix is a matrix whose rows and columns are the nodes of the cluster graph and whose entries are 0's or 1's. A 1 entry indicates that the corresponding nodes are connected, and a 0 indocates that they are not.

Usage

```
clust2Mat(memb)
```

Arguments

`memb`	A numeric vector, with each entry representing a node, the entry's value being the number of the cluster to which that node belongs.

Details

Given a vector of cluster membership numbers, where the number of entries is the number of nodes n, the function computes an nxn "adjacency matrix" for the corresponding cluster graph. The cluster graph is the graph in which two nodes are connected by an edge if and only if they are members of the same cluster. The adjacency matrix for the graph has rows and columns representing the nodes, in the same order as the input vector. The (i,j) entry is 1 if and only if node i and node j are in the same cluster. Otherwise, the entry is 0. By convention, diagonal entries are 0.

Value

An nxn adjacency matrix for the cluster graph, where n=length of cluster membership input vector memb.

Author(s)

Tom LaFramboise <tlaframb@hsph.harvard.edu>

See Also

makeClustM

Examples

```
memberships<-c(1,2,3,1,2,3,1,2,3,4)
clust2Mat(memberships)
```

---

depthmat *Matrices of depth of association for pairs of YEAST genes with respect to each of the BP,CC and MF ontologies of the GO database*

Description

This matrix of depths is used to obtain the predictome data in the paper. This is a symmetric matrix, where the i,j element corresponds to is the maximum depth of all annotations shared by genes i and j. Note that depth of a term in a specific Gene Ontology (BP, CC, MF) is defined as the shortest path between the term and the root node, where distance between nodes is measured by the number of edges traversed. Row labels of the matrix can be obtained by the row.names() function

Usage

```
data(depthmatBP)
```

Format

Each of three matrices, namely depthmatBP.rda, depthmatCC.rda, depthmatMF.rda is a symmetric matrix whose rows and columns correspond to specific YEAST genes (see row labels using row.names()). The i,j entry of each matrix refers to the maximum depth shared by genes i and j under each of the BP, CC and MF ontlogies respectively

Source

http://www.geneontology.org

Examples

```
data(depthmatBP)
print(row.names(depthmatBP)[1:10])
```

`getpvalue` *Function to obtain P values from the Edge permutation and Node permutation tests respectively*

Description

The function takes as inputs two adjacency matrices. Let X denote the observed number of edges in common between the two adjacency matrices. To test the significance of the correlation between the two data sources, the function performs N random edge permutations and random node permutations respectively. For each permutation test, the function outputs the proportion of N realizations that resulted in X edges or more at the intersection of the two datasources

Usage

```
getpvalue(act.mat, nonact.mat, num.iterations = 1000)
```

Arguments

`act.mat`	Adjacency matrix corresponding to first data source. That is, the i,j element of this matrix is 1 if data source one specifies a functional link between genes i and j
`nonact.mat`	Adjacency matrix corresponding to first data source. That is, the i,j element of this matrix is 1 if data source two specifies a functional link between genes i and j
`num.iterations`	Number of realizations from random edge (node) permutation to be obtained

Details

We note that the first adjacency matrix, denoted act.mat is the data source that is permutated with respect to edges or notes

Value

A vector of length 2, where the first element is the P value from Random Edge Permutation and the second element is the P value from Random Node Permutation

Author(s)

Raji Balasubramanian <rbalasub@hsph.harvard.edu>

See Also

`permEdgesM2M`, `permNodesM2M`, `makeClustM`

Examples

```
act.mat <- matrix(0,3,3)
act.mat[2,1] <- 1
act.mat[3,1] <- 1
nonact.mat  <- matrix(0,3,3)
nonact.mat[2,1] <- 1
nonact.mat[3,2] <- 1
p.val <- getpvalue(act.mat, nonact.mat, num.iterations = 100)
print(p.val)
```

`giaever` *Yeast Gene-Knockout Fitness Data*

Description

This data set contains fitness deficiency scores from gene knockout experiments involving yeast grown under a variety of altered environments (e.g. acid, heat, sorbitol, etc.)

Usage

```
data(giaever)
```

Format

A matrix whose rows are the 5922 genes knocked out and whose columns are the 32 experimental conditions.

Source

http://gobi.lbl.gov/YeastFitnessData

References

Giaever, G. et al. 2002 "Functional profiling of the Saccharomyces cerevisiae genome." Nature **418**, 387–391.

Examples

```
data(giaever)

## Find the 3000 most variable genes, according to sd/mean:
varMeas <- function(vec, na.rm=TRUE)
{
    if(na.rm)
      vec <- vec[!is.na(vec)]
    if(length(vec) == 0)
      measure <- NA
    else
      measure <- sd(vec)/mean(vec)
    return(measure)
}

variability <- apply(giaever, 1, varMeas)
```

```
rks <- rank(variability)

giaever3000 <- giaever[rks>length(rownames(giaever))-3000,]
```

---

mRNAclusters *Yeast mRNA Expression Data Cluster Memberships*

Description

This data set contains cluster membership for yeast genes clustered using mRNA expression from a microarray experiment in Causton, et al. Molecular Biology of the Cell (2001). The 3000 most variable genes were clustered using k-means with 30 clusters.

Usage

```
data(mRNAclusters)
```

Format

A data frame whose rows are the 3000 genes and whose two columns are gene name and cluster membership number.

Source

http://web.wi.mit.edu/young/environment

References

Causton, H. C. et al. 2001 "Remodeling of Yeast Genome Expression in Response to Environmental Changes." Molecular Biology of the Cell **12**, 323–337.

Examples

```
data(mRNAclusters)

## Compute the adjacency matrix for the corresponding cluster graph:
mRNAMat<-clust2Mat(mRNAclusters[,2])
```

---

makeClustM *Make an adjacency matrix for a cluster graph*

Description

This function takes a vector of cluster sizes and returns an adjacency matrix for a graph in which edges connect nodes if they are members of the same cluster.

Usage

```
makeClustM(nvec)
```

Arguments

nvec	A vector of cluster sizes

Value

A square adjacency matrix with the number of rows and columns equal to the sum of nvec. An entry of "1" in the ith row and jth column indicates that node i and node j are members of the same cluster. All other entries are "0".

Author(s)

Denise Scholtens

References

Balasubramanian, et al. (2004) A graph theoretic approach to integromics - integrating disparate sources of functional genomics data.

See Also

clust2Mat

Examples

```
a <- makeClustM(c(2,3,4))
```

---

mat2UndirG *Change graph representations*

Description

A function to turn an adjacency matrix for a graph into a graphNEL object.

Usage

```
mat2UndirG(V, mat)
```

Arguments

V	A vector of node names
mat	A square symmetric matrix indicating the presence of edges

Details

mat is a square matrix with rows and columns corresponding to nodes in the graph. Entries of "0" indicate the lack of an edge. Since this is making an undirected graph, mat must be symmetric.

Value

A graphNEL object.

Author(s)

Denise Scholtens

References

Balasubramanian, et al. (2004) A graph theoretic approach to integromics - integrating disparate sources of functional genomics data.

Examples

```
library(graph)
a <- matrix(c(0,1,1,1,1,0,0,0,1,0,0,0,1,0,0,0),ncol=4)
ag <- mat2UndirG(V=letters[1:4],mat=a)
```

---

permPower — *Function to compute estimated probability of detecting preferential connection of intracluster nodes*

Description

This function simulates graphs from the alternative hypothesis of preferential connection of intracluster nodes. For each graph, it runs a node and edge permutation test. The estimated "power" of each test is the proportion of graphs that the test rejects the null hypothesis of no preferential connection of intracluster edges.

Usage

```
permPower(psi=1,clsizes, nedge, nhyper=100, nperms=1000)
```

Arguments

psi	The non-centrality parameter for the noncentral hypergeometric distribution used to simulate the graphs.
clsizes	A vector of cluster sizes.
nedge	The number of edges in each graph.
nhyper	The number of noncentral hypergeometric graphs simulated to estimate "power".
nperms	The number of permutations used for each run of the edge and node permutation tests.

Details

The function first generates nhyper realizations of a noncentral hypergeometric(nedge,n,k,psi) random variable, where n is the number of node pairs and k is the number of intracluster node pairs. For each realization x, a graph with n edges, x of which are intracluster, is generated. The edge and node permutation tests (with nperms permutations each) are performed on each graph. The estimated "power" of each test is the proportion of graphs for which the test rejects the null hypothesis of no preferential connection of intracluster nodes (at the 5% level). The 95% confidence intervals for the power levels are also computed.

Value

A list with four components:

`power.permedge`
: Estimated "power" for edge permutation test.

`power.permnode`
: Estimated "power" for node permutation test.

`CI.permedge` Vector giving 95% confidence interval for edge permutation test power.

`CI.permnode` Vector giving 95% confidence interval for node permutation test power.

Author(s)

Tom LaFramboise <tlaframb@hsph.harvard.edu>

See Also

permEdgesM2M, permNodesM2M, makeClustM

Examples

```
permPower(psi=5,clsizes=c(1,2,3,4),nedge=10,nhyper=100,nperms=100)
```

---

perms *Randomly permute edges or node labels in a graph*

Description

Given an adjacency matrix for a graph, `permEdgesM2M` will return an adjacency matrix after an Erdos-Renyi random permutation of the edges in the graph. `perNodesM2M` will return an adjacency matrix for a graph with identical structure, but with the node labels permuted.

Usage

```
permEdgesM2M(mat)
permNodesM2M(mat)
```

Arguments

`mat` A square adjacency matrix for a graph.

Value

A square adjacency matrix for the new graph, subject to a random permutation of the edges or nodes.

Author(s)

Denise Scholtens

References

Balasubramanian, et al. (2004) A graph theoretic approach to integromics - integrating disparate sources of functional genomics data.

See Also

permPower

Examples

```
g <- matrix(c(0,1,1,1,1,0,0,0,1,0,0,0,1,0,0,0),nrow=4)

g1 <- permEdgesM2M(g)
g2 <- permNodesM2M(g)
```

Index