Package 'hierBipartite'

February 16, 2021

Type Package

Title Bipartite Graph-Based Hierarchical Clustering

Version 0.0.2

Maintainer Calvin Chi <calvin.chi@berkeley.edu>

Description Bipartite graph-based hierarchical clustering, developed for pharmacogenomic datasets and datasets sharing the same data structure. The goal is to construct a hierarchical clustering of groups of samples based on association patterns between two sets of variables. In the context of pharmacogenomic datasets, the samples are cell lines, and the two sets of variables are typically expression levels and drug sensitivity values. For this method, sparse canonical correlation analysis from Lee, W., Lee, D., Lee, Y. and Pawitan, Y. (2011) <doi:10.2202/1544-6115.1638> is first applied to extract association patterns for each group of samples. Then, a nuclear norm-based dissimilarity measure is used to construct a dissimilarity matrix between groups based on the extracted associations. Finally, hierarchical clustering is applied.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports parallel, magrittr, irlba

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Calvin Chi [aut, cre, cph],
Woojoo Lee [ctb],
Donghwan Lee [ctb],
Youngjo Lee [ctb],
Yudi Pawitan [ctb]

Repository CRAN

Date/Publication 2021-02-16 09:40:02 UTC

R topics documented:

`constructBipartiteGraph` *Construct Bipartite Graph Edge Weight Matrix of Gene-drug Association Patterns*

Description

Constructs edge weight matrix B representing association between set of variables in mat1 and set of variables in mat2 (see paper).

Usage

```
constructBipartiteGraph(
  mat1,
  mat2,
  n_subsample = 1,
  subsampling_ratio = 1,
  parallel = FALSE,
  maxCores = 7
)
```

Arguments

`mat1`	an n x p matrix of variable set 1 (e.g. gene expression)
`mat2`	an n x q matrix of variable set 2 (e.g. drug sensitivity)
`n_subsample`	number of times to perform subsampling to generate B
`subsampling_ratio`	fraction of samples to subsample each time
`parallel`	boolean for whether to parallelize subsampling
`maxCores`	maximum number of cores to use (only applicable when parallel = TRUE)

Value

a p x q matrix of bipartite graph edge weights

Examples

```
# Extract bipartite edge weight matrix B for cell lines from the
# squamous cell carcinoma, esophagus group
data(ctrp2)

groups = ctrp2$groups
X = ctrp2$X
Y = ctrp2$Y

x = X[groups[["squamous_cell_carcinoma_esophagus"]], ]
y = Y[groups[["squamous_cell_carcinoma_esophagus"]], ]

# Extract bipartite edge weight matrix B with subsampling
## Not run:
B = constructBipartiteGraph(x, y, n_subsample = 100,
                            subsampling_ratio = 0.90,
                            parallel = TRUE, maxCores = 2)

## End(Not run)
```

---

ctrp2 *Processed Cancer Cell Line Encyclopedia (CCLE) and Cancer Therapeutics Response Portal (CTRP2) Dataset*

Description

Smaller test dataset version of the "CTRP2" carcinoma dataset in the paper. Specifically, only the top 1,000 transcripts by correlation with drug sensitivity are included instead of 5,000. Otherwise the dataset has been processed exactly as described in the paper. Note the expression dataset is provided by CCLE and the drug sensitivity dataset is provided by CTRP2, and the pharmacogenomic datasets in the paper are are referred to by the resource providing the sensitivity data. The cell lines are grouped by carcinoma subtype and primary site (e.g. lung NSC).

Usage

```
data(ctrp2)
```

Format

A list with elements of gene expression, drug sensitivities, and group membership.

X n x p gene expression matrix

Y n x q drug sensitivities matrix

groups List of starting groups. Each group is represented by a vector of row indices for X, Y.

References

Barretina, J., et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature, 483(7391), 603–607. (PubMed)

Seashore-Ludlow, B., et al. (2015). Harnessing connectivity in a large-scale small-molecule sensitivity dataset. Cancer discovery, 5(11), 1210-1223. (PubMed)

Examples

```
data(ctrp2)
X = ctrp2[["X"]]
Y = ctrp2[["Y"]]
groups = ctrp2[["groups"]]
```

`getSignificantMergedGroups` *Select Significant Results from 'HierBipartite'*

Description

Selects clusters from bipartite graph-based hierarchical clustering with p-value less than or equal to a p-value cutoff.

Usage

```
getSignificantMergedGroups(results, p = 0.05)
```

Arguments

`results`	list of results from bipartite graph-based hierarchical clustering
`p`	p-value cutoff

Value

list of results from bipartite graph-based hierarchical clustering, but only with clusters with p-value at or below p-value cutoff

Examples

```
# sample bipartite graph-based hierarchical clustering of three groups
data(ctrp2)

groups = ctrp2$groups
X = ctrp2$X
Y = ctrp2$Y

groupNames = names(groups)
groupSmall = groups[groupNames[1:3]]
```

```
## Not run:
result = hierBipartite(X, Y, groupSmall)

# set fictitious p-values, with one cluster with p-value less than the cutoff
# and the other not
result$nodePvals = list(0.03, 0.12)
getSignificantMergedGroups(result, p = 0.05)

## End(Not run)
```

`hierBipartite` *Bipartite Graph-based Hierarchical Clustering*

Description

Main bipartite graph-based hierarchial clustering algorithm. Visit here for vignette on using the hierBipartite package.

Usage

```
hierBipartite(
  X,
  Y,
  groups,
  link = "ward.D2",
  n_subsample = 1,
  subsampling_ratio = 1,
  p.value = FALSE,
  n_perm = 100,
  parallel = FALSE,
  maxCores = 7,
  p_cutoff = 0.1
)
```

Arguments

`X`	an n x p matrix of variable set 1 (e.g. gene expression)
`Y`	an n x q matrix of variable set 2 (e.g. drug sensitivity)
`groups`	a list of starting group membership (e.g. list("1" = c(1,2,3), "2" = c(4,5,6)) means group 1 has samples 1, 2, 3, and group 2 has samples 4, 5, 6.
`link`	string indicating link function as input to hclust(). One of "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", "centroid".
`n_subsample`	number of subsampling to generate matrix B (see paper)
`subsampling_ratio`	fraction of samples to sample for subsampling to generate matrix B (see paper)

p.value	boolean for whether to generate p-values for each merge
n_perm	number of permutations for generating p-values. Ignored if p.value = FALSE
parallel	boolean for whether to parallelize subsampling and p-value generation step
maxCores	maximum number of cores to use (only applicable when parallel = TRUE)
p_cutoff	p-value cutoff that determines whether merge is significant. If p-value > p_cutoff, p-values will not be calculated for future merges involving current group. Ignored if p.value = FALSE.

Value

list of results from bipartite graph-based hierarchical clustering, containing up to

- hclustObj: hclust object
- groupMerges: list of clusters after each merge, in order of merge. Each cluster is indicated by a vector of cell line groups
- nodePvals: list of p-value of each new merge, in order of merge. Only available if p.value = TRUE
- D: dissimilarity matrix

Examples

```
# Get a small subset of the test dataset
data(ctrp2)

groups = ctrp2$groups
X = ctrp2$X
Y = ctrp2$Y

groupNames = names(groups)
groupSmall = groups[groupNames[1:3]]

## Not run:
# Basic call of hierBipartite() on small test dataset
result0 = hierBipartite(X, Y, groupSmall)

# Calling hierBipartite() with subsampling
result1 = hierBipartite(X, Y, groupSmall, n_subsample = 100, subsampling_ratio = 0.90)

# Calling hierBipartite() with p-value generation
result2 = hierBipartite(X, Y, groupSmall, n_perm = 100, p.value = TRUE, p_cutoff = 0.10)

# Calling hierBipartite() with both subsampling and p-value generation (expensive)
result3 = hierBipartite(X, Y, groupSmall, n_subsample = 100, subsampling_ratio = 0.90,
                        n_perm = 100, p.value = TRUE, p_cutoff = 0.10)

## End(Not run)
```

`matrixDissimilarity` *Matrix dissimilarity*

Description

Computes nuclear norm-based dissimilarity measure between two matrices.

Usage

```
matrixDissimilarity(B1, B2)
```

Arguments

B1	first p x q bipartite graph edge weight matrix
B2	second p x q bipartite graph edge weight matrix

Value

nuclear norm-based dissimilarity

Examples

```
# Compute matrix dissimilarity in edge weight matrix between squamous cell
# carcinoma, esophagus and squamous cell carcinoma, upper aerodigestive
data(ctrp2)

groups = ctrp2$groups
X = ctrp2$X
Y = ctrp2$Y

x1 = X[groups[["squamous_cell_carcinoma_esophagus"]], ]
y1 = Y[groups[["squamous_cell_carcinoma_esophagus"]], ]

## Not run:
B1 = constructBipartiteGraph(x1, y1)

## End(Not run)

x2 = X[groups[["squamous_cell_carcinoma_upper_aerodigestive"]], ]
y2 = Y[groups[["squamous_cell_carcinoma_upper_aerodigestive"]], ]

## Not run:
B2 = constructBipartiteGraph(x2, y2)
matrixDissimilarity(B1, B2)

## End(Not run)
```

null_distri *Null distribution of dissimilarity measures*

Description

Generates null distribution of dissimilarity measures between group 1 (X1, Y1) and group 2 (X2, Y2).

Usage

```
null_distri(X1, Y1, X2, Y2, n.perm = 100, parallel = FALSE, maxCores = 7)
```

Arguments

X1	an n x p matrix of variable set 1 (e.g. gene expression) from group 1
Y1	an n x q matrix of variable set 2 (e.g. drug sensitivity) from group 1
X2	an n x p matrix of variable set 1 (e.g. gene expression) from group 2
Y2	an n x q matrix of varaible set 2 (e.g. drug sensitivity) from group 2
n.perm	number of null dissimilarity measures to generate
parallel	boolean for whether to parallelize permutation
maxCores	maximum number of cores to use (only applicable when parallel = TRUE)

Value

vector of length n.perm of null dissimilarity measures

Examples

```
# Get data for group squamous cell carcinoma, esophagus and for group
# squamous cell carcinoma, upper aerodigestive
data(ctrp2)

groups = ctrp2$groups
X = ctrp2$X
Y = ctrp2$Y

x1 = X[groups[["squamous_cell_carcinoma_esophagus"]], ]
y1 = Y[groups[["squamous_cell_carcinoma_esophagus"]], ]

x2 = X[groups[["squamous_cell_carcinoma_upper_aerodigestive"]], ]
y2 = Y[groups[["squamous_cell_carcinoma_upper_aerodigestive"]], ]

## Not run:
dissimilarities = null_distri(x1, y1, x2, y2, n.perm = 100)

## End(Not run)
```

p_value *P-value of Similarity in Gene-drug Associations*

Description

Computes p-value as number of null dissimilarities less than or equal to observed dissimilarity.

Usage

```
p_value(dissimilarity, dissimilarities)
```

Arguments

dissimilarity	observed dissimilarity
dissimilarities	null distribution of dissimilarities

Value

p-value

Examples

```
# simulate null distribution of dissimilarities
dissimilarities = runif(100, min = 0, max = 1)

d = 0.10
p_value(d, dissimilarities)
```

scca *Sparse canonical covariance analysis*

Description

'scca' is used to perform sparse canonical covariance analysis (SCCA)

Usage

```
scca(X,Y,penalty="HL",lamx=c(1,2,3),lamy=c(1,2,3),nc=1,
tuning="CV.alt",K=5,seed=NULL,center=TRUE,scale=FALSE)
```

Arguments

X	n-by-p data matrix, where n is the number of subjects and p is the number of variables
Y	n-by-q data matrix, where q is the number of variables
penalty	"HL" is the unbounded penalty proposed by Lee and Oh (2009). "LASSO" (Tibshirani, 1996), "SCAD" (Fan and Li, 2001) and "SOFT" (soft thresholding) are also available as other penalty options. Default is "HL".
lamx	A vector specifying grid points of the tuning parameter for X. Default is (1,2,3).
lamy	A vector specifying grid points of the tuning parameter for Y. Default is (1,2,3).
nc	Number of components (canonical vectors). Default is 1.
tuning	How to find optimal tuning parameters for the sparsity. If tuning="CV.full", then the tuning parameters are selected automatically via K-fold cross-validation by using 2-dim'l grid search. If "CV.alt", then a sequential 1-dim'l search method is applied instead of the 2-dim'l grid search. Default is "CV.alt".
K	Perform K-fold cross-validation.
seed	Seed number for initialization. A random initial point is generated for tuning="CV.alt".
center	The columns of the data matrix are centered to have mean zero. Default is TRUE.
scale	The columns of the data matrix are scaled to have variance 1. Default is FALSE.

Details

Sparse CCA uses a random-effect model approach to obtain sparse regression. This model gives unbounded gains for zero loadings at the origin. Various penalty functions can be adapted as well.

Value

- A: p-by-nc matrix, k-th colum of A corresponds to k-th pattern
- B: q-by-nc matrix, k-th colum of B corresponds to k-th pattern (canonical vector) for Y
- U: n-by-nc matrix. k-th column of U corresponds to k-th score associated with k-th pattern for X
- V: n-by-nc matrix. k-th column of V corresponds to k-th score associated with k-th pattern for Y
- lambda: nc-by-2 matrix. k-th row of lambda corresponds to the optimal tuning parameters for k-th pattern pairs
- CR: average cross-validated sample covariance

Author(s)

Woojoo Lee, Donghwan Lee, Youngjo Lee and Yudi Pawitan

References

Lee, W., Lee, D., Lee, Y. and Pawitan, Y. (2011) Sparse Canonical Covariance Analysis for High-throughput Data

Examples

```
## Example 1
## A very simple simulation example
n<-10; p<-50; q<-20
X = matrix(rnorm(n*p),ncol=p)
Y = matrix(rnorm(n*q),ncol=q)
scca(X,Y)
```

Index