Package 'MetaPCA'

February 19, 2015

Type Package

Title MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data

Version 0.1.4

Author Don Kang <donkang75@gmail.com> and George Tseng <ctseng@pitt.edu>

Maintainer Don Kang <donkang75@gmail.com>

Description MetaPCA implements simultaneous dimension reduction using PCA when multiple studies are combined. We propose two basic ideas to find a common PC subspace by eigenvalue maximization approach and angle minimization approach, and we extend the concept to incorporate Robust PCA and Sparse PCA in the meta-analysis realm.

Depends R (>= 2.10.0), foreach

Suggests MASS, GEOquery, pcaPP, affy, hgu133plus2.db, doMC, doSMP, ellipse, impute

License GPL-2

URL https://github.com/donkang75/MetaPCA

LazyLoad yes

Collate MetaPCA.R functions.R DropDupGenes.R requireAll.R PlotPC2D.R PreprocessMetaAnalysis.R

Date 2011-06-15

Repository CRAN

Date/Publication 2011-06-15 18:49:55

NeedsCompilation no

R topics documented:

`MetaPCA-package` *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

MetaPCA implements simultaneous dimension reduction using PCA when multiple studies are combined. We propose two basic ideas to find a common PC subspace by eigenvalue maximization approach and angle minimization approach, and we extend the concept to incorporate Robust PCA and Sparse PCA in the meta-analysis realm.

Details

Package:	MetaPCA
Type:	Package
Version:	0.1.4
Date:	2011-06-15
License:	GPL-2
LazyLoad:	yes

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

References

Dongwan D. Kang and George C. Tseng. (2011) Meta-PCA: Meta-analysis in the Dimension Reduction of Genomic data.

Examples

```
## Not run:
#Spellman, 1998 Yeast cell cycle data set
#Consider each synchronization method as a separate data
data(Spellman)
pc <- list(alpha=prcomp(t(Spellman$alpha))$x, cdc15=prcomp(t(Spellman$cdc15))$x,
cdc28=prcomp(t(Spellman$cdc28))$x, elu=prcomp(t(Spellman$elu))$x)
#There are currently 4 meta-pca methods. Run either one of following four.
metaPC <- MetaPCA(Spellman, method=”Eigen”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”Angle”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”RobustAngle”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”SparseAngle”, doPreprocess=FALSE)
#Comparing between usual pca and meta-pca
#The first lows are four data sets based on usual PCA, and
#the second rows are by MetaPCA
#We're looking for a cyclic pattern.
par(mfrow=c(2,4), cex=1, mar=c(0.2,0.2,0.2,0.2))
for(i in 1:4) {
plot(pc[[i]][,1], pc[[i]][,2], type=”n”, xlab=””, ylab=””, xaxt=”n”, yaxt=”n”)
text(pc[[i]][,1], pc[[i]][,2], 1:nrow(pc[[i]]), cex=1.5)
lines(pc[[i]][,1], pc[[i]][,2])
}
for(i in 1:4) {
plot(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2], type=”n”, xlab=””, ylab=””, xaxt=”n”, yaxt=”n”)
text(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2], 1:nrow(metaPC$x[[i]]$coord), cex=1.5)
lines(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2])
}

#4 prostate cancer data which have three classes: normal, primary, metastasis
data(prostate)
#There are currently 4 meta-pca methods. Run either one of following four.
metaPC <- MetaPCA(prostate, method=”Eigen”, doPreprocess=FALSE, .scale=TRUE)
metaPC <- MetaPCA(prostate, method=”Angle”, doPreprocess=FALSE)
metaPC <- MetaPCA(prostate, method=”RobustAngle”, doPreprocess=FALSE)
metaPC <- MetaPCA(prostate, method=”SparseAngle”, doPreprocess=FALSE)
#Plotting 4 data in the same space!
coord <- foreach(dd=iter(metaPC$x), .combine=rbind) %do% dd$coord
PlotPC2D(coord[,1:2], drawEllipse=F, dataset.name=”Prostate”, .class.order=c(”Metastasis”,”Primary”,”Normal”),
.class.color=c('red','#838383','blue'), .annotation=T, newPlot=T,
.class2=rep(names(metaPC$x), times=sapply(metaPC$x,function(x)nrow(x$coord))),
.class2.order=names(metaPC$x), .points.size=1)

#In the case of ”SparseAngle” method, the top contributing genes for all studies can be determined
#For instance, top 20 genes in 1st PC and their coefficients
metaPC$v[order(abs(metaPC$v[,1]), decreasing=TRUE),1][1:20]


## End(Not run)
```

DropDupGenes *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

When multiple probesets share the same gene symbols, select only the best probeset in terms of IQR

Usage

```
DropDupGenes(dat, isParallel=FALSE, nCores=NULL, na.rm=TRUE)
```

Arguments

`dat`	A gene expression matrix which has genes in rows and samples in columns.
`isParallel`	Whether to use multiple cores in parallel for fast computing. By default, it is false.
`nCores`	When isParallel is true, the number of cores can be set. By default, all cores in the machine are used in the unix-like machine, and 3 cores are used in windows.
`na.rm`	Whether to remove genes which have no annotation. Default is TRUE.

Value

A gene expression matrix which has unique genes in rows and samples in columns.

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

References

Dongwan D. Kang and George C. Tseng. (2011) Meta-PCA: Meta-analysis in the Dimension Reduction of Genomic data.

Examples

```
## Not run:
#One of example that shows how to generate a expression matrix used in the analysis
requireAll(c('GEOquery', 'affy', 'hgu133plus2.db'))
#It might be needed to download the source files first, and save it to local directory
#such as "./data/Prostate/Varambally" in this example
#ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SeriesMatrix/GSE3325/GSE3325_series_matrix.txt.gz
Varambally <- getGEO('GSE3325', destdir="./data/Prostate/Varambally")
Varambally <- Varambally[[1]]
Varambally.sLabel <- as.character(pData(Varambally)$title)
Varambally.sLabel[grep("Benign",Varambally.sLabel)] <- "Normal"
Varambally.sLabel[grep("primary",Varambally.sLabel)] <- "Primary"
Varambally.sLabel[grep("Metastatic",Varambally.sLabel)] <- "Metastasis"
Varambally <- exprs(Varambally)
colnames(Varambally) <- Varambally.sLabel
rownames(Varambally) <- unlist(mget(rownames(Varambally), hgu133plus2SYMBOL))
Varambally <- DropDupGenes(Varambally, na.rm=TRUE)
Varambally <- log2(Varambally)

## End(Not run)
```

`MetaPCA` *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

MetaPCA implements simultaneous dimension reduction using PCA when multiple studies are combined. We propose two basic ideas to find a common PC subspace by eigenvalue maximization approach and angle minimization approach, and we extend the concept to incorporate Robust PCA and Sparse PCA in the meta-analysis realm.

Usage

```
MetaPCA(DList, method=c("Angle","Eigen","RobustAngle","SparseAngle"), robust.var=c("qn","mad"), nPC=
.weight=rep(1/length(DList),length(DList)), sparse.maxFeatures=NULL, sparse.lambda=NULL,
sparse.max.iter=100, sparse.eps=1e-3, .scale=FALSE, .scaleAdjust=TRUE, doPreprocess=TRUE,
cutRatioByMean=.4, cutRatioByVar=.4, doImpute=TRUE,na.rm.pct=.1, na.rm.pct.each=.5,
verbose=FALSE)
```

Arguments

`DList` A list of all data matrices; Each data name should be set as the name of each list element. Each data should be a numeric matrix that has genes in the rows and samples in the columns. Row names should be official gene symbols and column names be sample labels.

`method` A vector of four meta PCA methods. The first two methods are basic approaches; the last two are extended approaches of robust PCA and sparse PCA but may be rather slower than the basic methods. Default is "Angle", which is angle minimization method. See the details in the reference.

`robust.var` Robust measure of variance when "RobustAngle" method was selected in the method.

`nPC` The number of returned PC's, i.e. the number of dimension reduced by PCA.

`.weight` Weight for each data if information is available. Default is equal weight.

`sparse.maxFeatures`
The number of genes left for the Sparse PCA approach. If NULL (default), it is determined based on the default lambda.

`sparse.lambda` The parameter lambda which determines the sparsity of loading vectors. The default is calculated as the number of data divided by square root of the number of overall genes.

`sparse.max.iter`
The number of maximum iteration for achieving convergence of sparse loading vectors. Default is 100.

`sparse.eps` The convergence decision precision level. Default is 1e-3.

`.scale` Whether to apply gene based normalization. Default is FALSE. But for the "Eigen" method, gene scaling is recommended for the comparability reason of covariance matrix.

`.scaleAdjust` Whether to apply scaling adjustment for a comparable visualization. Default is TRUE.

`doPreprocess` Whether to apply gene filtering. Default is TRUE. However "SparseAngle" method do not use gene filtering.

`cutRatioByMean` Proportion of genes filtered by study-wise mean. Default is 40%.

`cutRatioByVar` Proportion of genes filtered by study-wise variance. Default is 40%.

`doImpute` Whether to impute missing genes. Default is TRUE, and default imputation method is knn.

`na.rm.pct` Proportion of genes filtered by study-wise missing proportion. Default is 10%.

`na.rm.pct.each` Proportion of genes filtered by each study’s missing proportion. Default is 50%.

`verbose` Whether to print logs. Default is FALSE.

Value

list object having the specified number of PC’s of all data sets and loading matrix of meta subspace.

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

References

Dongwan D. Kang and George C. Tseng. (2011) Meta-PCA: Meta-analysis in the Dimension Reduction of Genomic data.

Examples

```
## Not run:
#Spellman, 1998 Yeast cell cycle data set
#Consider each synchronization method as a separate data
data(Spellman)
pc <- list(alpha=prcomp(t(Spellman$alpha))$x, cdc15=prcomp(t(Spellman$cdc15))$x,
cdc28=prcomp(t(Spellman$cdc28))$x, elu=prcomp(t(Spellman$elu))$x)
#There are currently 4 meta-pca methods. Run either one of following four.
metaPC <- MetaPCA(Spellman, method=”Eigen”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”Angle”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”RobustAngle”, doPreprocess=FALSE)
metaPC <- MetaPCA(Spellman, method=”SparseAngle”, doPreprocess=FALSE)
#Comparing between usual pca and meta-pca
#The first lows are four data sets based on usual PCA, and
#the second rows are by MetaPCA
#We're looking for a cyclic pattern.
par(mfrow=c(2,4), cex=1, mar=c(0.2,0.2,0.2,0.2))
for(i in 1:4) {
plot(pc[[i]][,1], pc[[i]][,2], type=”n”, xlab=””, ylab=””, xaxt=”n”, yaxt=”n”)
text(pc[[i]][,1], pc[[i]][,2], 1:nrow(pc[[i]]), cex=1.5)
lines(pc[[i]][,1], pc[[i]][,2])
}
for(i in 1:4) {
plot(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2], type="n", xlab="", ylab="", xaxt="n", yaxt="n")
text(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2], 1:nrow(metaPC$x[[i]]$coord), cex=1.5)
lines(metaPC$x[[i]]$coord[,1], metaPC$x[[i]]$coord[,2])
}

#4 prostate cancer data which have three classes: normal, primary, metastasis
data(prostate)
#There are currently 4 meta-pca methods. Run either one of following four.
metaPC <- MetaPCA(prostate, method="Eigen", doPreprocess=FALSE, .scale=TRUE)
metaPC <- MetaPCA(prostate, method="Angle", doPreprocess=FALSE)
metaPC <- MetaPCA(prostate, method="RobustAngle", doPreprocess=FALSE)
metaPC <- MetaPCA(prostate, method="SparseAngle", doPreprocess=FALSE)
#Plotting 4 data in the same space!
coord <- foreach(dd=iter(metaPC$x), .combine=rbind) %do% dd$coord
PlotPC2D(coord[,1:2], drawEllipse=F, dataset.name="Prostate", .class.order=c("Metastasis","Primary","Normal"),
.class.color=c('red','#838383','blue'), .annotation=T, newPlot=T,
.class2=rep(names(metaPC$x), times=sapply(metaPC$x,function(x)nrow(x$coord))),
.class2.order=names(metaPC$x), .points.size=1)

#In the case of "SparseAngle" method, the top contributing genes for all studies can be determined
#For instance, top 20 genes in 1st PC and their coefficients
metaPC$v[order(abs(metaPC$v[,1]), decreasing=TRUE),1][1:20]


## End(Not run)
```

PlotPC2D *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

2D PCA plots.

Usage

```
PlotPC2D(coord, drawObjects=TRUE, drawEllipse=FALSE, dataset.name=NULL,
pctInfo=NULL, main=NULL, sub=NULL, xlab=NULL, ylab=NULL, newPlot=TRUE,
.points.size=1, .class=rownames(coord), .class.order=NULL, .class.color=NULL,
.class2=NULL, .class2.order=NULL, .class2.shape=NULL, .annotation=TRUE,
.legend=c("bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "cent
```

Arguments

`coord` 2D Coordinates matrix of objects. Rows are objects and columns are coordinates.

drawObjects Whether to draw objects as points.

drawEllipse Whether to draw ellipses estimated from objects 2D distribution.

dataset.name Name to be displayed as a part of title.

pctInfo Explained percentage of variance by each PC.

main Main title.

sub Sub title.

xlab Label for x-axis.

ylab Label for y-axis.

newPlot Whether to draw a plot in the new frame.

.points.size Size of objects' points.

.class Object's class label such as disease classification.

.class.order The order of class representation.

.class.color The color of class representation.

.class2 The second class label of each object such as study name.

.class2.order The order of 2nd class representation.

.class2.shape The shape of 2nd class representation.

.annotation Whether to present annotation such as x,y axis labels, legend, or titles.

.legend Location of legend in a plot.

Value

NA. A PCA plot is drawn.

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

References

Dongwan D. Kang and George C. Tseng. (2011) Meta-PCA: Meta-analysis in the Dimension Reduction of Genomic data.

Examples

```
## Not run:
#4 prostate cancer data which have three classes: normal, primary, metastasis
data(prostate)
metaPC <- MetaPCA(prostate, method="Angle", doPreprocess=FALSE)
#Plotting 4 data in the same space with ellipses overlayed!
coord <- foreach(dd=iter(metaPC$x), .combine=rbind) %do% dd$coord
PlotPC2D(coord[,1:2], drawEllipse=T, dataset.name="Prostate", .class.order=c("Metastasis","Primary","Normal"),
.class.color=c('red','#838383','blue'), .annotation=T, newPlot=T,
.class2=rep(names(metaPC$x), times=sapply(metaPC$x,function(x)nrow(x$coord))),
.class2.order=names(metaPC$x), .points.size=1)


## End(Not run)
```

`PreprocessMetaAnalysis` *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

Preprocessing for microarray meta-analysis. It is about gene filtering and missing value imputation.

Usage

```
PreprocessMetaAnalysis(DList, cutRatioByMean=.4, cutRatioByVar=.4, doImpute=FALSE, na.rm.pct=.1, na.
```

Arguments

`DList`	A list of all data matrices; Each data name should be set as the name of each list element. Each data should be a numeric matrix that has genes in the rows and samples in the columns. Row names should be official gene symbols and column names be sample labels.
`cutRatioByMean`	Proportion of genes filtered by study-wise mean. Default is 40%.
`cutRatioByVar`	Proportion of genes filtered by study-wise variance. Default is 40%.
`doImpute`	Whether to impute missing genes. Default is TRUE, and default imputation method is knn.
`na.rm.pct`	Proportion of genes filtered by study-wise missing proportion. Default is 10%.
`na.rm.pct.each`	Proportion of genes filtered by each study's missing proportion. Default is 50%.
`verbose`	Whether to print logs. Default is FALSE.

Value

list object of all data matrices after filtering and imputation.

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

References

Dongwan D. Kang and George C. Tseng. (2011) Meta-PCA: Meta-analysis in the Dimension Reduction of Genomic data.

Examples

```
## Not run:
DList <- PreprocessMetaAnalysis(list(Yu=Yu, Lapointe=Lapointe, Tomlins=Tomlins, Varambally=Varambally),
cutRatioByMean=.1, cutRatioByVar=.1, doImpute=T, na.rm.pct=.2)
str(DList)

## End(Not run)
```

prostate *4 prostate cancer studies*

Description

4 prostate cancer studies comparing three classes: normal, primary, metastasis.

Data Name	Published Year	Array Platform	Sample Size	GEO Accession ID
Lapointe	2004	cDNA	112	GSE3933
Yu	2004	HG-U95Av2	108	GSE6919
Varambally	2005	HG-U133 Plus 2	19	GSE3325
Tomlins	2007	cDNA	76	GSE6099

Usage

```
prostate
```

Format

A list containing 4 matrices. Each matrix is gene expression data after gene filtering.

Source

Gene Expression Omnibus (GEO)

References

Lapointe, J., Li, C., Higgins, J. P., Van De Rijn, M., Bair, E., Montgomery, K., Ferrari, M., Egevad, L., Rayford, W., Bergerheim, U. et al. (2004). Gene expression profiling identifies clinically relevant subtypes of prostate cancer. Proceedings of the National Academy of Sciences of the United States of America 101 811.

Yu, Y. P., Landsittel, D., Jing, L., Nelson, J., Ren, B., Liu, L., McDonald, C., Thomas, R., Dhir, R., Finkelstein, S. et al. (2004). Gene expression alterations in prostate cancer predicting tumor aggression and preceding development of malignancy. Journal of Clinical Oncology 22 2790.

Varambally, S., Yu, J., Laxman, B., Rhodes, D. R., Mehra, R., Tomlins, S. A., Shah, R. B., Chandran, U., Monzon, F. A., Becich, M. J. et al. (2005). Integrative genomic and proteomic analysis of prostate cancer reveals signatures of metastatic progression. Cancer cell 8 393-406.

Tomlins, S. A., Mehra, R., Rhodes, D. R., Cao, X., Wang, L., Dhanasekaran, S. M., Kalyana-Sundaram, S., Wei, J. T., Rubin, M. A., Pienta, K. J. et al. (2006). Integrative molecular concept modeling of prostate cancer progression. Nature genetics 39 41-51.

requireAll *MetaPCA: Meta-analysis in the Dimension Reduction of Genomic data*

Description

requireAll description

Usage

```
requireAll(packages)
```

Arguments

`packages`	A character vector of required packages. Unavailable packages are going to be installed.

Value

None

Author(s)

Don Kang (donkang75@gmail.com) and George Tseng (ctseng@pitt.edu)

Examples

```
## Not run:
libs <- c("proto", "foreach", ifelse(.Platform$OS.type == "unix", "doMC", "doSMP"))
requireAll(libs)

## End(Not run)
```

Spellman *4 Spellman cancer studies*

Description

Yeast cell-cycle data set was divided into four subsets which correspond to the four different synchronization methods: alpha arrest (alpha), arrest of cdc15 or cdc28 temperature-sensitive mutant (cdc15 and cdc28), and elutriation (elu). We filtered out genes which have overall missing values >= 10% or log2 transformed standard deviation >= .45. 1025 genes were left, and the number of time points in the experiments were 18, 24, 17, and 14 for alpha, cdc15, cdc28, and elu, respectively. Additionally, we have imputed missing values using knn.

Usage

```
Spellman
```

Format

A list containing 4 matrices. Each matrix is gene expression data after gene filtering.

Source

Gene Expression Omnibus (GEO)

References

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell 9 3273.

Index