Package 'multiDimBio'

October 13, 2022

Type Package

Title Multivariate Analysis and Visualization for Biological Data

Version 1.2.2

Date 2020-04-09

Author Samuel V. Scarpino

Maintainer Samuel V. Scarpino <scarpino@utexas.edu>

Description Code to support a systems biology research program from inception through publication. The methods focus on dimension reduction approaches to detect patterns in complex, multivariate experimental data and places an emphasis on informative visualizations. The goal for this project is to create a package that will evolve over time, thereby remaining relevant and reflective of current methods and techniques. As a result, we encourage suggested additions to the package, both methodological and graphical.

License GPL (>= 3.0)

LazyLoad yes

Imports ggplot2 (>= 3.2.0), lme4 (>= 1.1-21), pcaMethods (>= 1.76.0), misc3d (>= 0.8-4), MASS (>= 7.3-29), RColorBrewer(>= 1.1-2), gridGraphics (>= 0.1-5)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-09 17:10:06 UTC

R topics documented:

`multiDimBio-package` *Multivariate Analysis and Visualization for Biological Data*

Description

Code to support a systems biology research program from inception through publication. The methods focus on dimension reduction approaches to detect patterns in complex, multivariate experimental data and places an emphasis on informative visualizations. The goal for this project is to create a package that will evolve over time, thereby remaining relevant and reflective of current methods and techniques. As a result, we encourage suggested additions to the package, both methodological and graphical.

Details

Package:	multiDimBio
Type:	Package
Version:	1.2.2
Date:	2020-04-09
License:	GPL 3.0
LazyLoad:	yes

The datasets are: Nuclei, Groups, CondA, CondB, Scores, and Dyad

The main functions are: boxWhisker, completeData, F_select, intPlot, ldaPlot, loadings, meanCent, percentMax, permuteLDA, power, ppca_mdb, zTrans, binomPower, h2Estimate, and plotBinomPower.

Type ?<object> to learn more about these objects, e.g. ?Nuclei

Type ?<function> to see examples of the function's use, e.g. ?FSelect

Author(s)

Samuel V Scarpino Maintainer: Samuel V Scarpino <scarpino@utexas.edu>

References

Collyer M, Adams D. (2007) Analysis of Two - State Multivariate Phenotypic Change in Ecological Studies. Ecology: 88(3) 683 - 692.

Costanza M, Afifi A. (1979) Comparison of Stopping Rules in Forward Stepwise Discriminant Analysis. Journal of the American Statistical Association: pp. 777 - 78

Crews D, Gillette R, Scarpino SV, Manikkam M, Savenkova MI, Skinner MK. (2012) Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA: 109(23) 9143 - 9148.

Davies SW, Scarpino SV, Pongwarin T, Scott J, Matz MV. (2015) Estimating Trait Heritability in Highly Fecund Species. G3: Genes| Genomes| Genetics: 5(12) 2639 - 45.

Habbema J, Hermans J. (1977) Selection of Variables in Discriminant Analysis by F-Statistics and Error Rate. Technometrics: 19(4) 487 - 493.

Jennrich R. (1977) Stepwise discriminant analysis, volume 3. New York Wiley Sons.

Roweis S. (1997) EM algorithms for PCA and sensible PCA. Neural Inf. Proc. Syst.: 10 626 - 632.

Stacklies W, Redestig H, Scholz M, Walther D, Selbig J. (2007) pcaMethods - a Bioconductor package providing PCA methods for incomplete data. Bioinformatics: 23 1164 - 1167.

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R. (2001) Missing value estimation methods for DNA microarrays. Bioinformatics: 17(6) 520 - 5252.

See Also

pcaMethods

`binomPower` *Power analysis for estimating the heritability of a binomial trait*

Description

Performs a power analysis for estimating the heritability of a binomial trait. This function can take a long time to run if either nsims or nperms is large.

Usage

```
binomPower(ndads, mm, vv, tau2, nperms, nsims, nbins, alpha = 0.05, doPlot=FALSE)
```

Arguments

ndads	a (non-empty) numeric value indicating the number of dads.
mm	a (non-empty) numeric value indicating the mean number of offspring per dad per bin (normal dist). mm must be less than vv.
vv	a (non-empty) numeric value indicating the variance in offspring per dad per bin (normal dist). vv. must be greater than mm.
tau2	a (non-empty) numeric value indicating the dad effect (narrow-sense heritability ~ tau2/(tau2+(pi/sqrt(3))^2)).
nperms	a (non-empty) numeric value indicating the number of bootstrap permutations to use for caluclating a p value.
nsims	a (non-empty) numeric value indicating the number of simulations to run per parameter combination.
nbins	a (non-empty) numeric value indicating the number of bins, data are pooled before analysis.
alpha	a (non-empty) numeric value indicating the cutoff for significant p values.
doPlot	a (non-empty) logical value indicating whether to plot the results of the power analysis.

Value

Returns a list and an optional set of .pdfs (if doPlot==TRUE). The list contains:

roc	a data.frame with the summarized results of the power analysis.
params	a numeric matrix with the paramater values.
results	a numeric matrix with the full results of the analysis.

Examples

```
ndads <- c(9,18)
mm <- 4.629634
vv <- 6.31339
tau2 <- c(0,0.5)
nperms <- 2
nsims <- 2
nbins <- 3
doPlot <- TRUE
binomPower(ndads,mm,vv,tau2,nperms,nsims,nbins,doPlot)
```

boxWhisker *A function to create a box and whisker plot by group ID*

Description

A function to create a box and whisker plot by group ID.

Usage

```
boxWhisker(data, groups, palette = "Paired")
```

Arguments

data	a (non-empty) matrix of data values
groups	a (non-empty) vector of group IDs with length equal to the number of rows in data
palette	A color palette for plotting. The default is 'Paired.' See colorbrewer2.org for alternatives.

Value

Returns a box-whisker plot of the data by group ID.

Examples

```
data(Nuclei)
data(Groups)
boxWhisker(Nuclei, Groups)

#changing the color palette

boxWhisker(data = Nuclei, groups = Groups, palette = 'Set1')
```

completeData *Function to impute missing data.*

Description

This function imputes missing data using a probabilistic principle component analysis framework and is a wrapper around functions implemented in the pcaMethods package (Stacklies et al. 2007), was proposed by Troyanskaya et al 2001 and is based on methods developed in Roweis 1997.

Usage

```
completeData(data, n_pcs, cut.trait = 0.5, cut.ind = 0.5, show.test = TRUE)
```

Arguments

data	a (non-empty) numeric matrix of data values.
n_pcs	a (non-empty) numeric value indicating the desired number of principle component axes.
cut.trait	a number indicating the maximum proportion of missing traits before an individual is removed from data. A value of 1 will not remove any individuals and 0 will remove them all.
cut.ind	a number indicating the maximum proportion of individuals missing a trait score before that trait is removed from data. A value of 1 will not remove any traits and 0 will remove them all.
show.test	a logical statement indicating whether a diagnostic plot of the data imputation should be returned.

Value

Returns a list with two entries.

complete_dat	an object of class matrix with missing values imputed using a probabilistic principle component framework.
plots	a list of plots stored as grid plots.

References

Roweis S (1997). EM algorithms for PCA and sensible PCA. Neural Inf. Proc. Syst., 10, 626 - 632.

Stacklies W, Redestig H, Scholz M, Walther D, Selbig J (2007). pcaMethods - a Bioconductor package providing PCA methods for incomplete data. Bioinformatics, 23, 1164 - 1167.

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R (2001). Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6), 520 - 5252.

See Also

pcaMethods, pca

Examples

```
data(Nuclei)
npcs<-floor(ncol(Nuclei)/5)

length(which(is.na(Nuclei))==TRUE)

dat.comp<-completeData(data = Nuclei, n_pcs = npcs)

length(which(is.na(dat.comp))==TRUE)
```

CondA *Treatment condition for animals contained in the data set Nuclei*

Description

Animals measured in the Nuclei data set were either from linneages exposed to the fungicide Vinclozolin (Vinclozolin) or not (Control).

Usage

```
data(CondA)
```

Format

A factor vector indicating which treatment group the individuals in Nuclei belong to.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(CondA)
```

CondB *Stress condition for animals contained in the data set Nuclei*

Description

Animals measured in the Nuclei data set were either subjected to chronic restraint stress (stress) or not (control).

Usage

```
data(CondB)
```

Format

A factor vector indicating which stress group the individuals in Nuclei belong to.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(CondB)
```

Dyad *Housing dyad for animals contained in the data set Nuclei*

Description

Animals measured in the Nuclei data set were housed in dyads with one individual from the Vinclozolin line and one from the control line housed together. Each dyad was either stressed or not stressed.

Usage

```
data(Dyad)
```

Format

A factor vector indicating which housing dyad the individuals in Nuclei are in.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(Dyad)
```

FSelect *A Function to perform step-wise discriminant analysis using F statistics*

Description

Select data using a F tests

Usage

```
FSelect(Data, Group, target, p.adj.method = "holm", Missing.Data = "Complete")
```

Arguments

Data	A (non-empty), numeric matrix of data values
Group	A (non-empty), vector indicating group membership. Length(unique(Group))==2
target	The number of desired traits. Target cannot be greater than the number of columns in Data
p.adj.method	The method used to control for false discovery. The default setting is 'holm'
Missing.Data	The method used to handle missing data. The default, 'Complete' will use completeData to impute missing data, setting Missing.Data='Remove' will remove all individuals with missing data. FSelect cannot handle missing data.

Value

FSelect returns list containing at least the following components:

Selected	An ordered list indicating which columns were selected.
F.Selected	An ordered list containing the F statistics for each column indicated in Selected.
PrF	An ordered list containing the p values for each column indicated in Selected.
PrNotes	A string indicating which method was used to control for multiple comparisons
model	An lm object with the final model results.

References

Costanza M, Afifi A (1979). Comparison of Stopping Rules in Forward Stepwise Discriminant Analysis. Journal of the American Statistical Association, pp. 777 - 78

Habbema J, Hermans J (1977). Selection of Variables in Discriminant Analysis by F - Statistics and Error Rate. Technometrics, 19(4), 487 - 493.

Jennrich R (1977). Stepwise discriminant analysis, volume 3. New York Wiley Sons.

See Also

completeData

Examples

```
data(Nuclei)
data(Groups)
npcs<-floor(ncol(Nuclei)/5)

dat.comp <- completeData(data = Nuclei, n_pcs = npcs)
groups.use <- c(1,2)
use.dat <- which(Groups==groups.use[1]|Groups==groups.use[2])

dat.use <- Nuclei[use.dat,]
GR.use <- Groups[use.dat]

#not run
#FSelect(DAT.use,GR.use,3)
```

`getP` *An internal function for getting empirical p values*

Description

Simulates p values.

Usage

```
getP(ndads, mm, vv, tau2, nperms, nsims, nbins)
```

Arguments

`ndads`	a (non-empty) numeric value indicating the number of dads.
`mm`	a (non-empty) numeric value indicating the mean number of offspring per dad per bin (normal dist). mm must be less than vv.
`vv`	a (non-empty) numeric value indicating the variance in offspring per dad per bin (normal dist). vv must be greater than mm.
`tau2`	a (non-empty) numeric value indicating the dad effect (narrow-sense heritability ~ tau2/(tau2+(pi/sqrt(3))^2)).
`nperms`	a (non-empty) numeric value indicating the number of bootstrap permutations to use for caluclating a p value.
`nsims`	a (non-empty) numeric value indicating the number of simulations to run per parameter combination.
`nbins`	a (non-empty) numeric value indicating the number of bins, data are pooled before analysis.

Value

Returns a vector of simulated p values. The list contains:

Examples

```
ndads <- c(9,18)
mm <- 4.629634
vv <- 6.31339
tau2 <- c(0,0.5)
nperms <- 2
nsims <- 2
nbins <- 3
getP(ndads, mm, vv, tau2, nperms, nsims, nbins)
```

Groups *The group ID for animals contained in the data set Nuclei*

Description

Animals measured in the Nuclei data set belong to one of four groups determined by their linneage (Vinclozolin or Control) and their stress treatment (Stressed or Non-Stressed).

Usage

```
data(Groups)
```

Format

A factor vector indicating which group the individuals in Nuclei are in.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(Groups)
```

`h2Estimate` *Estimates the heritability of a binomial trait*

Description

Estimates the narrow-sense heritability of a binomial trait and calculates a p value by randomization.

Usage

```
h2Estimate(data,nreps=1000)
```

Arguments

`data`	a (non-empty) numeric matrix with three columns. The first two should contain the trait data (number of occurances of each outcome type) and the third should contain the group ids.
`nreps`	a (non-empty) numeric value indicating the number of resamples to perform when calculating the emperical p value.

Details

Estimates the narrow-sense heritability of a binomial trait. This function works by fitting two models, one with and one without a random-effect of sire. These models are compared by randomizing the sire ids nreps times and re-fitting the model. For each of the nreps model pairs, a deviance is calculated and a "p value" estimated by comparing that distribution of deviance to the observed. The heritability is approximatly tau2/(tau2+(pi/sqrt(3))^2), where tau2 is the random-effect variance due to sire.

Value

Returns a list. The list contains:

`h2`	The estimated narrow-sense heritability. The narrow-sense heritability is approximatly tau2/(tau2+(pi/sqrt(3))^2), where tau2 is the random-effect variance due to sire.
`pval`	The probability that the best-fit model includes an extra variance term for sire (random effect of dad). The value is calculated by comparing the deviances from nreps number of randomized model comparisions.
`deviance`	The deviance between a null model without a random effect of dad and a model with.
`sim`	The simulated deviances used in calculating the p value in pval.
`obsMod`	The glmer model object resulting from the observed data.

Examples

```
#non-zero heritability
ndads <- 18
mm <- 4
vv <-  6
tau2 <- 2.5
nbins <- 3

mylogit <- function(x) log(x/{1-x})
ilogit <- function(x) 1/{1+exp(-x)}

swimprob <- ilogit(rnorm(ndads, 0, sqrt(tau2)))
mytable <- NULL
for(i in 1:ndads) {
bincounts <- pmax(1,rnbinom(nbins, mu = mm, size = mm^2/{vv-mm}))
swim <- rbinom(3, bincounts,swimprob[i])
set <- bincounts - swim
theserows <- data.frame(set=set,swim=swim, Dad = i, Bin = 1:nbins)
mytable <- rbind(mytable, theserows)
}

est <- h2Estimate(mytable,nreps=10)

print(est$h2)

#zero heritability
ndads <- 18
mm <- 4
vv <-  6
tau2 <- 0
nbins <- 3

mylogit <- function(x) log(x/{1-x})
ilogit <- function(x) 1/{1+exp(-x)}

swimprob <- ilogit(rnorm(ndads, 0, sqrt(tau2)))
mytable0 <- NULL
for(i in 1:ndads) {
bincounts <- pmax(1,rnbinom(nbins, mu = mm, size = mm^2/{vv-mm}))
swim <- rbinom(3, bincounts,swimprob[i])
set <- bincounts - swim
theserows <- data.frame(set=set,swim=swim, Dad = i, Bin = 1:nbins)
mytable0 <- rbind(mytable0, theserows)
}

est0 <- h2Estimate(mytable0,nreps=10)

print(est0$h2)
```

`IntPlot` *A function to visualize the results of a MANOVA*

Description

The function produces an interaction plot to demonstrate the results of a MANOVA using the function interaction.plot.

Usage

```
IntPlot(Scores, Cov.A, Cov.B, pvalues = rep(1, 8), int.pvalues = rep(1, 4))
```

Arguments

Scores	A (non-empty) numeric matrix of principle component scores or raw data.
Cov.A	A (non-empty) bivariate factor vector indicating the factor for each row in Scores
Cov.B	A (non-empty) bivariate factor vector indicating the factor for each row in Scores
pvalues	An optional vector of p values for each covariate across Scores. The length of pvalues must equal the number of columns in Scores times 2.
int.pvalues	An optional vector of p values for each interaction. The length of int.pvalues must equal the number of columns in Scores.

Value

a list of plots stored as grid plots.

See Also

interaction.plot

Examples

```
data(Scores)
data(CondA)
data(CondB)

pvals<-c(0.03,0.6,0.05,0.07,0.9,0.2,0.5,0.3)
int.pvals<-c(0.3,0.45,0.5,0.12)

IntPlot(Scores,CondA,CondB,pvalues=pvals, int.pvalues=int.pvals)
```

LandscapePlot *A function to visualize the Functional Landscape of measured traits*

Description

This function plots a three-dimensional landscape of measured traits. The peak heights are relative with respect to the input data. The width of each peak is controlled by the argument sigma and has only an aesthetic purpose. The 3D image is generated using the drawScene and surfaceTriangles .

Usage

```
LandscapePlot(Data, Groups=NULL, PDF=FALSE,LocPlot=FALSE,control=c(75,1,30))
```

Arguments

Data	A (non-empty) numeric matrix with trait values
Groups	A (non-empty)factor vector indicating the group membership of each row in Data. If there is only a single group present in Data then Groups=NULL (default).
PDF	Logical controlling whether to output the results as a .pdf or a .jpeg. The default (PDF=FALSE) will produce a .jpeg. The file size for .pdf output can be large.
LocPlot	Logical controlling whether to output a .pdf naming the peaks according to the columns they represent. The defaul is FALSE.
control	An optional numeric vector setting the control parameters for persp. control[1] = theta, control[2] = r, control[3] = phi

Value

a list of plots stored as grid plots (or.pdf if PDF=TRUE) file for each column in data.

Examples

```
data(Nuclei)
data(Groups)

#plotting the first six columns
#not run
#LandscapePlot(Nuclei[,1:6], Groups=Groups)
```

---

ldaPlot *A function to visualize the results of a discriminant analysis*

Description

The function takes as input the traits and group IDs and will perform a discriminate function analysis and visualize the results. For the pair-wise comparison of groups we use density histograms with points along the x-axis denoting the actual data, Figure 3 For multi-group comparisons we plot a bivariate scatter for all pairwise combinations of discriminate axes. The color of plotting symbols can be altered using the palette argument and the axes comparisons (with max n = number of groups - 1).

Usage

```
ldaPlot(Data, Groups, palette = "BrBG", axes = c(1, 2, 2, 3, 1, 3))
```

Arguments

Data	A (non-empty), numeric matrix of data values
Groups	A (non-empty), vector indicating group membership. Length(unique(Group))==2
palette	A color palette for plotting. The default is 'Paired.' See colorbrewer2.org for alternatives.
axes	A numeric vector describing which axes to compare. For example, axes=c(1,2) will on produce a single plot comparing the first and second axis.

Value

Returns a list of ggplot2 plots.

See Also

lda

Examples

```
data(Nuclei)
data(Groups)
ldaPlot(Nuclei, Groups, palette='BrBG', axes=c(1,2,2,3,1,3))
```

Loadings *A function to visualize trait loadings onto discriminant function and principle component axes*

Description

This function produces barplots representative of the contribution of a particular trait or variable to either a discriminant function or principle component axis.

Usage

```
Loadings(DATA, GROUPS, method = c("PCA", "LDA"))
```

Arguments

DATA	A (non-empty) numeric matrix with trait values
GROUPS	A (non-empty)factor vector indicating the group membership of each row in DATA
method	An optional list indicating whether the results for a principle component analysis, 'PCA', or linear discriminant analysis, 'LDA' should be performed.

Value

Outputs a list with values and plots for each test listed in method.

See Also

pca, lda

Examples

```
data(Nuclei)
data(Groups)
Loadings(Nuclei, Groups, method=c("PCA", "LDA"))
```

---

makeCompMat *A function to create a pairwise comparison matrix*

Description

This function creates a pairwise comparison matrix for n groups. All possible pairwise combinations are created, with rows in the matrix equal to the desired comparison.

Usage

```
makeCompMat(ng)
```

Arguments

ng	A single number indicating the total number of unique groups

Value

Returns a matrix with two columns and ng choose 2 rows.

See Also

PermuteLDA

Examples

```
makeCompMat(3)

makeCompMat(4)

data(Groups)
NGroups<-length(unique(Groups))

makeCompMat(NGroups)
```

MeanCent *A function to scale data to mean 0*

Description

This function rescales the columns in a data matrix to have mean 0. The variance is not scaled and missing values are ignored in the calculation.

Usage

```
MeanCent(DATA)
```

Arguments

DATA	A (non-empty) matrix with data values. Columns should be different traits and rows unique observations of those traits

Value

Returns a matrix with the same dimensions as DATA.

See Also

ZTrans, PercentMax

Examples

```
data(Nuclei)

colMeans(Nuclei, na.rm=TRUE)

Nuclei.MC<-MeanCent(Nuclei)

colMeans(Nuclei.MC, na.rm=TRUE)
```

Nuclei *Brain activity in 14 brain regions for 71 individuals*

Description

The activity in 14 brain nuclei were measured in rats that were in one of four groups: 1) Non-stressed, Control 2) Stressed, Control 3) Non-stressed, Vinclozolin 4) Stressed, Vinclozolin

Usage

```
data(Nuclei)
```

Format

A numeric matrix with 71 individuals as rows and the activity of 14 brain nuclei as columns. NAs indicate missing data.

Details

Two different cohorts of male rats of the F3 generation of Vinclozolin (Vinclozolin-Lineage) and Vehicle Control (Control-Lineage) Lineages produced at Washington State University are shipped to the University of Texas on the day after weaning. Rats are randomly pair-housed (one Control-Lineage and one Vinclozolin-Lineage animal) and remain in these dyads throughout the duration of the study. Half of the dyads are randomly chosen to receive chronic restraint stress (CRS) treatment for 6 hours daily for 21 consecutive days commencing 1 hr after lights off. Activity in 14 brain nuclei were measured at the end of the study.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(Nuclei)
```

---

partialF *A function to compute partial F statistics*

Description

This is an internal function used in FSelect. It can only be used for two groups. The partial F statistic is the additional contribution to the model from adding one more trait.

Usage

```
partialF(m.lda, GROUP, T_pm1)
```

Arguments

m.lda	An object of class 'lda'
GROUP	A factor vector indicating group membership
T_pm1	The F statistic calculated for a discriminant analysis with only the most informative trait.

Value

Returns a partial F statistic

References

Habbema J, Hermans J (1977). Selection of Variables in Discriminant Analysis by F-Statistics and Error Rate. Technometrics, 19(4), 487 - 493.

See Also

FSelect

Examples

```
#Internal function used in FSelect

data(Nuclei)
data(Groups)

NPC<-floor(ncol(Nuclei)/5)

DAT.comp<-completeData(Nuclei, n_pcs = NPC)
Groups.use<-c(1,2)
use.DAT<-which(Groups==Groups.use[1]|Groups==Groups.use[2])

DAT.use<-Nuclei[use.DAT,]
GR.use<-Groups[use.DAT]

traitA<-2

mlda<-MASS::lda(GR.use~DAT.use[,traitA])

F1<-partialF(mlda,GR.use,0)

traitB<-1

mlda2<-MASS::lda(GR.use~DAT.use[,c(traitA,traitB)])

partialF(mlda2,GR.use,F1)
```

PercentMax *A function to scale data to the percent of the maximum observed*

Description

This function rescales the columns in a data matrix to the percent of the maximum observed value. The variance is not scaled and missing values are ignored in the calculation.

Usage

```
PercentMax(DATA)
```

Arguments

DATA	A (non-empty) matrix with data values. Columns should be different traits and rows unique observations of those traits

Value

Returns a matrix with the same dimensions as DATA.

See Also

ZTrans, MeanCent

Examples

```
data(Nuclei)

colMeans(Nuclei, na.rm=TRUE)

Nuclei.PM<-PercentMax(Nuclei)

colMeans(Nuclei.PM, na.rm=TRUE)
```

PermuteLDA *A function to determine whether two groups are in statistically different locations in multivariate space See Collyer and Adams 2007*

Description

The function calculates the multivariate distance between two groups across all traits and determines whether they differ signifcantly using a Monte Carlo randomization test. The Monte Carlo randomization creates a null distribution by randomizing the residual deviation from the group mean across all individuals. This method controls for heteroscedasticity and was designed by Collyer and Adams (2007) for use in analyzing data sets that have sparse groups sizes relative to the number of traits.

Usage

```
PermuteLDA(Data, Groups, NPerm, Missing.Data = "Complete")
```

Arguments

Data	A (non-empty), numeric matrix of data values
Groups	A (non-empty), vector indicating group membership.
NPerm	The number of permutations used to generate the null distribution. The default is 100.
Missing.Data	The method used to handle missing data. The default, 'Complete' will use CompleteData to impute missing data, setting Missing.Data='Remove' will remove all individuals with missing data. FSelect cannot handle missing data.

Details

Determining the statistical significance of a discriminate function analysis along with performing that analysis on sparse data sets, e.g. many traits observed on comparatively few individuals, is a challenge. Collyer and Adams (2007) developed a Monte Carlo based algorithm for addressing both of those issues. Briefly, the test uses the underlying Var/Cov structure of the data and randomizes the group membership to calculate a null distribution. This test simultaneously controls for heteroscedasticity, a common problem in sparse data sets and allows the approximation of a p-value for the test. For the original implementation and formulation of the method see Collyer and Adams (2007) or http://www.public.iastate. edu/~dcadams/software.html. Unlike the FSelect implementation, PermuteLDA will work properly with an arbitrary number of groups. The time required to run the algorithm is non-linear in the number of groups.

Value

Returns a data frame with four columns and the number of groups choose 2 rows. Each row is a pairwise comparison between groups. The column 'Pr' is the p value to reject the null hypothesis of no difference (a value in 'Pr' < 0.05 would result in rejecting the hypothesis that the two groups are not different. The column 'Distance' is the multivariate distance between the two groups.

References

Collyer M, Adams D (2007). Analysis of Two - State Multivariate Phenotypic Change in Ecological Studies. Ecology, 88(3), 683 - 692.

For an implementation of the original method coded in R see http://www.public.iastate. edu/~dcadams/software.html.

See Also

PermuteLDA

Examples

```
data(Nuclei)
data(Groups)
PermuteLDA(Nuclei,Groups,50)
```

plotBinomPower *A function to plot the results of a binomPower run*

Description

A function to plot the results of a binomPower run.

Usage

```
plotBinomPower(datPlotBig,params)
```

Arguments

datPlotBig	a (non-empty) matrix of data values, with columns trueTau, ndads, trueTau2
params	a (non-empty) matrix of parameter values, with columns mm and vv.

Value

Returns a list of two plots of the binomPower analysis results.

Examples

```
#not run
```

Power *A function to estimate the error rate for FSelect and PermuteLDA.*

Description

Methods are implemented to compute the statistical power, in terms of the type II error rate, based on anticipated sample and effect sizes for FSelect() and PermuteLDA(). By default the power of both tests are determined by iterating over a range of effect and sample sizes. The default settings were selected to be representative of many behavioral genetic studies; however, users can input alternative sample and effect sizes. For high values of trials this function can be very slow.

Usage

```
Power(func = "PermuteLDA", N = "DEFAULT.N", effect.size = "DEFAULT.e", trials = 100)
```

Arguments

func	A character string indicating which function to compute the power for, can be either 'PermuteLDA' or 'FSelect'
N	A (non-empty) vector of group sizes. The lenght of N must be greater than 1 and tha minimum group size for 'FSelect' can not be less than 6. The size of each group is N/2.
effect.size	A (non-empty) vector or single value of effect sizes.
trials	A number indicating the number of trials for each combination of N and effect.size to calculate the power.

Details

The algorithm for the power analysis proceeds as follows: 1. Input sample and effect sizes 2. Set the number of significant effects, e to 0. Note - Total number of traits is fixed at 6 3. Draw random deviates for the given sample size for 6 traits. Note - All traits not significant under this iteration are drawn from a N(0,1) distribution. 4. Perform either FSelect() or PermuteLDA() and record the results. 5. Return to step 3 N times, recording the results each time. Note - N is set using the trials input 6. If e<5 return to step 2 and set the number of significant effects to e+1 7. Proceed to the next combination of sample and effect size. 8. Output the results for each combination of sample and effect size as a function of the number of significant traits.

Value

Outputs a list with plots and results for each effect size.

See Also

PermuteLDA,FSelect

Examples

```
#not run
#Power(func = 'FSelect', N=c(6,8), effect.size=0.5, trials = 2)
```

PPCA *A function to perform a probabilistic principle component analysis*

Description

Performs a probabilistic principle component analysis using the function 'pca' in the package'pcaMethods'

Usage

```
PPCA(Data, nPCs=4, CENTER=TRUE, SCALE='vector')
```

Arguments

Data	A (non-empty), numeric matrix of data values
nPCs	The number of resulting principle component axes. nPCs must be less than or equal to the number of columns in Data.
CENTER	A logical statement indicating whether data should be centered to mean 0, TRUE, or not, FALSE.
SCALE	A character string indicating which method should be used to scale the variances. The default setting is 'vector.'

Details

In PPCA an Expectation Maximization (EM) algorithm is used to fit a Gaussian latent variable model (Tippping and Bishop (1999)). A latent variable model seeks to relate an observed vector of data to a lower dimensional vector of latent (or unobserved) variables, an approach similar to a factor analysis. Our implementation is a wrapper around the pcaMethods functions ppca and svdimpute (Stacklies et al. (2007)) and is included mainly for convience. The method used in pca was adapted from Roweis (1997) and a Matlab script developed by Jakob Verbeek.

Value

Returns an object of class 'pcaRes.' See documentation in the package codepcaMethods

References

Roweis S (1997). EM algorithms for PCA and sensible PCA. Neural Inf. Proc. Syst., 10, 626 - 632.

Stacklies W, Redestig H, Scholz M, Walther D, Selbig J (2007). pcaMethods - a Bioconductor package providing PCA methods for incomplete data. Bioinformatics, 23, 1164 - 1167.

Tippping M, Bishop C (1999). Probabilistic Principle Componenet Analysis. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 61(3), 611 - 622.

See Also

pcaMethods, pca

Examples

```
data(Nuclei)
PPCA1<-PPCA(Nuclei, nPCs=2, CENTER=TRUE, SCALE='vector')
Scores1<-PPCA1@scores
```

Scores — *Principle component scores based on the data in Nuclei*

Description

Principle component scores were computed using PPCA for the data set Nuclei.

Usage

```
data(Scores)
```

Format

A numeric matrix with 4 columns and the same number of rows as Nuclei. There are no missing values.

Source

The data are provided courtesy of David Crews at the University of Texas at Austin.

References

Crews, D, R Gillette, SV Scarpino, M Manikkam, MI Savenkova, MK Skinner. 2012. Epigenetic Transgenerational Alterations to Stress Response in Brain Gene Networks and Behavior. Proc. Natl. Acad. Sci. USA. 109 (23). 9143 - 9148.

Examples

```
data(Scores)

data(Nuclei)

SCORES<-PPCA(Nuclei)@scores
```

simPower — *An internal function of binomPower, which actually calculates the p value*

Description

An internal function of binomPower, which actually calculates the p value.

Usage

```
simPower(ndads,mm,vv,tau2,nperms,nbins)
```

Arguments

ndads	a (non-empty) numeric value indicating the number of dads.
mm	a (non-empty) numeric value indicating the mean number of offspring per dad per bin (normal dist). mm must be less than vv.
vv	a (non-empty) numeric value indicating the variance in offspring per dad per bin (normal dist). vv must be great than mm.
tau2	a (non-empty) numeric value indicating the dad effect (narrow-sense heritability ~ tau2/(tau2+(pi/sqrt(3))^2)).
nperms	a (non-empty) numeric value indicating the number of bootstrap permutations to use for caluclating a p value.
nbins	a (non-empty) numeric value indicating the number of bins, data are pooled before analysis.

Value

Returns a p value for a given set of conditions over a specificed number of bootstrap permutations.

Examples

```
#not run
```

ZTrans *A function to convert data into a z-score*

Description

This function converts the columns in a data matrix into z-scores. The score is computed by subracting each observation in a column from the column mean and divding by the column standard deviation. Each column is converted independently of the others missing values are ignored in the calculation.

Usage

```
ZTrans(DATA)
```

Arguments

DATA	A (non-empty) matrix with data values. Columns should be different traits and rows unique observations of those traits

Value

Returns a matrix with the same dimensions as DATA.

See Also

PercentMax, MeanCent

Examples

```
data(Nuclei)

colMeans(Nuclei, na.rm=TRUE)

Nuclei.ZT<-ZTrans(Nuclei)

colMeans(Nuclei.ZT, na.rm=TRUE)
```

Index