Package 'chroGPS'

March 26, 2013

Type Package

Title chroGPS: navigating through the epigenome

Version 1.0.1

Date 2012-09-30

Author Oscar Reina, David Rossell

Maintainer Oscar Reina <oscar.reina@irbbarcelona.org>

Description We provide intuitive maps to visualize the association between genetic elements, with emphasis on epigenetics. The approach is based on Multi-Dimensional Scaling. We provide several sensible distance metrics, and adjustment procedures to remove systematic biases typically observed when merging data obtained under different technologies or genetic backgrounds.

License GPL (>=2)

Depends R (>= 2.13.0), IRanges, methods, Biobase, MASS, graphics,stats, rgl

Imports graphics, cluster, DPpackage

Enhances multicore

Collate adjustPeaks.R distGPS.R domainDist.R mds-class.R mds.R procrustesAdj.R clusGPS.R geneSetGPS.R getmodEncode.R gff2RDList.R

LazyLoad yes

biocViews Visualization, Clustering

R topics documented:

---

addVar *Plot vector of a quantitative variable over a MDS map.*

Description

Given a quantitative variable as a numeric vector with one element for each point on a MDS map, calculate and plot the weight vector corresponding to that variable.

Usage

```
addVar(mds1, z, plot = TRUE, label = "z", pos = 3, ...)
```

Arguments

mds1	An object of class mds with the MDS object.
z	Numeric vector with the quantitative variable, one element for each point.
plot	Set to TRUE to calculate and draw the resulting vector on the MDS.
label	Something to be printed on the tip of the vector arrow, usually the name of the given variable.
pos	Graphical position where the label is drawn respect to the vector arrow tip.
...	Additional parameters given to the generic function plot.

Value

A named list with the vector components.

Examples

```
# Not run
# See chroGPS-manual.pdf for examples.
```

adjustPeaks *Adjust peak width so that samples obtained under different conditions become comparable.*

Description

Peaks obtained under different conditions (e.g. chip-chip, chip-seq, mnase-seq) are typically not comparable in terms of their width. adjustPeaks modifies the mean and SD of the peak width distribution for each condition, so that they become equivalent to the condition with widest peaks. See details.

Usage

adjustPeaks(x, adjust, sampleid, logscale = TRUE)

Arguments

x	RangedDataList indicating the binding sites for each sample/experiment.
adjust	Vector indicating the adjustment factor, i.e. the condition under which each sample has been obtained.
sampleid	Vector containing the sample identifier. sampleid should take the same value for samples obtained under different conditions, as this is used to detect the samples to be used for Procrustes adjustment.
logscale	If set to TRUE the mean and SD are matched for log width, otherwise the original widths are used. Working in log scale can help reduce the effect of outliers (e.g. an usually long binding site).

Details

In a sense, the peak calling resolution is decreased so that they become comparable to the less precise technology (notice that there is no reliable way to increase the precision given by a low-resolution technology).

Value

RangedDataList object with adjusted widths.

Methods

signature(x='RangedDataList') Each element in x contains the binding sites for a different sample. The start, end and chrosomome of each binding sites should be accessed via start, end and space.

See Also

procrustesAdj for an alternative, more general, adjustment based on Procrustes. distGPS for computing distances, mds to create MDS-oriented objects.

Examples

```
#See examples in help(procrustesAdj)
```

boostMDS *Improve goodness-of-fit of a given MDS solution in terms of R-square.*

Description

Given a distance matrix and a valid MDS representation for it, improve the R-square correlation between observed and approximated distances until converged is reached for a given threshold.

Usage

boostMDS(D, Y, rate = 0.1, maxit = 50, tol = 0.001, samplesize = 1,
verbose = TRUE, scale = FALSE, seed = 149, plt = FALSE, mc.cores = 1)

Arguments

D	Distance matrix.
Y	Matrix with points from a valid MDS solution for the distances in D.
rate	Grid step rate, start with 0.1 which usually is a good compromise, try also 0.01, 1, 10.
maxit	Maximum number of iterations.
tol	Tolerace for R-square convergence.
samplesize	If a big MDS is given, point resampling can be done in order to speed up the process, use values in (0,1].
verbose	Give details of the gains in R-square and step size.
scale	Whether to scale the MDS coordinates in the output MDS.
seed	A random seed to be used in the resampling process if samplesize < 1.
plt	Whether to plot the intermediate solutions or not.
mc.cores	Number of cores to use in parallelized grid step size search.

Value

The function returns a matrix with the coordinates of a valid MDS solution for distance matrix D where the R-square correlation has been improved. However, have in mind that an MDS solution with better R-square does not necessarily mean the solution is easier to interpret. As with any MDS approach, a balance must be found between pure 'technical' goodness-of-fit and usefulness of the delivered solution in terms of answering the original hypothesis.

References

boostMDS is based on hitMDS (High-Throughput Multidimensional Scaling, see see http://dig.ipk-gatersleben.de/hitmds/hitmds.html for details)

Examples

```
# Not run, see also chroGPS-manual.pdf file for examples
#data(geneSample)
#d = distGPS(geneSample,uniqueRows=TRUE)
#m = mds(d,type='isoMDS')
#m
#plot(m)
#m = boostMDS(d@d,m@points)
#plot(m)
```

---

clusGPS — *Hierarchical clustering (Agglomerative Nesting) of elements based on their input pairwise distances.*

Description

Elements are clustered using Agglomerative Nesting of their pairwise dissimilarities (distances). Additionally, semiparametric bayesian density is estimated using a Dirichlet process mixture of normals. This is used both to compute bayesian mis-classification posterior probabilities and to estimate probability contours for each cluster in any given cut of the clustering tree.

contour2dDP, contour3dDP and plotContour functions can be used to compute and plot bayesian density probability contours for a given set of elements (points) from a pre-generated MDS object. These functions are used internally by clusGPS to draw cluster contours but are also useful to visualize other type of contours over the map (ie genes from a given Gene Ontology term, having a specific epigenetic mark of interest, etc).

Usage

```
clusGPS(d, m, h, grid, ngrid=1000, densgrid=TRUE, type = "agnes", method =
"average", samplesize = 1, p.adjust = TRUE, k = 2:5, mc.cores = 1,
set.seed = 149, verbose=TRUE, ...)
contour2dDP(x, ngrid, grid = NULL, probContour = 0.5, xlim, ylim,
   labels = "", labcex = 0.01, col = colors()[393], lwd = 4,
   lty = 1, contour.type = "single", contour.fill = FALSE, ...)
contour3dDP(x, col = "red", probContour = 0.5, ngrid = 30, contour.type = "none")
```

Arguments

d	Object of class distGPS with the pairwise observed dissimilarities between elements.
m	(Optional). Object of class mds with a MDS object generated from the distances in d. Only MDS type "boostMDS" is available. The mds function performs an optimization of the approximated distances in m in order to improve r-square correlation between them and the observed dissimilarities en d, maximizing goodness of fit.
h	(Optional). Object of class hclust with a pre-calculated clustering for the elements in d.
grid	Matrix of dimension ngrid*nvar giving the diagonal points of the grid where the density estimate is evaluated. The default value is NULL: grid dimensions are

chosen according to the range of the data, and granularity is automatically determined according to data density, in order to provide a more accurate estimation in high density areas, where more resolution is needed.

ngrid — Number of grid points where the density estimate is evaluated. This argument is ignored if a grid is specified. The default value is 1000. Higher values are recommended if data presents very high density areas.

densgrid — Set to true to generate grid points from the quantile distribution of the data using the grid size defined by ngrid. This is useful if the data presents areas of very different density, ranging from very sparse to extremely dense areas, optimizing grid granularity where is necessary, therefore improving resolution of density estimation and reducing computation time.

type — Type of clustering to be performed. Currently only "agnes" (Agglomerative Nesting) is supported, but any other clustering type can be used by providing a pre-calculated object h.

method — Clustering method. See agnes for details.

samplesize — Proportion of elements to sample for computing clustering and density estimation. This is useful to generate density contours from a subset of the data, speeding up computation.

p.adjust — Set to TRUE to adjust the bayesian posterior probabilities of mis-classification.

k — Integer vector indicating the number of clusters on which density estimation will be computed for mis-classification or contour calculation.

mc.cores — Number of cores to be used for parallel computation with the multicore package.

set.seed — If samplesize<1, random seed to be used to perform random sampling of the data.

verbose — Set to TRUE to output clustering process information.

x — Numeric matrix indicating coordinates of the points for which a probability contour is calculated in contour2dDP and contour3dDP.

probContour — Numeric matrix indicating coordinates of the points for which a probability contour is calculated in contour2dDP and contour3dDP.

contour.type — For contour2dDP, type of contour, either 'single' (surrounding the points within the given probContour probability) or 'multiple' to generate terrain-like density contour lines. For contour3dDP currently only 'contours' (to drawn contours as tridimensional polygons) and 'none' are available.

contour.fill — Deprecated.

xlim,ylim,labels,labcex,col,lwd,lty — Graphical parameters given to contour2dDP.

... — Additional parameters.

Value

Object of class clusGPS. See help for clusGPS-methods for details.

Methods

signature(d='distGPS',m='mds') Agglomerative nesting is performed for the elements whose pairwise distances are given in d. For each cluster partition given in k, cluster identity for each element is returned, and semiparametric bayesian density estimation is computed using the point density information from m.

signature(d='distGPS',m='missing') Agglomerative nesting is performed for the elements whose pairwise distances are given in d. For each cluster partition given in k, cluster identity for each element is returned.

plot signature(m = "clusGPS"): S4 plot method for mds objects.

Author(s)

Oscar Reina

Examples

```
# Not run
# data(geneSample)
# # Computing distances
# d <- distGPS(geneSample,metric='tanimoto',uniqueRows=TRUE)
# # Creating MDS
# mds1 <- mds(d)
# mds1
# plot(mds1)
# # Calculating densities (contours and probabilities), takes a while
# c=clusGPS(d,mds1,k=2:3,mc.cores=1)
# # c contains information for contours and probabilities
# plot(c,type='contours',k=3,lwd=3,probContour=.75)
# plot(c,type='stats',k=3)
# plot(c,type='avgstat',k=3)
# plot(c,type='density',k=3,ask=TRUE,xlim=range(mds1@points),ylim=range(mds1@points))
```

---

clusGPS-class *Class* "clusGPS"

Description

Agglomerative Nesting for a distGPS object. Contains probability contours and bayesian posterior probability of mis-classification for the clusters evaluated.

Details

Parameters for the S4 plot method for mds objects.

Object of class "mds" with a 2D or 3D Multidimensional Scaling to be plotted.

drawlabels**:** TRUE to use rownames of the MDS points as text labels.

labels**:** Alternative character vector giving the text labels for the MDS points.

plantar**:** If a 3D MDS is used, set plantar to TRUE to plot projected views of the MDS using XY, YZ and XZ axis decomposition.

point.cex**:** Size of the points / spheres for the MDS plot.

text.cex**:** Size of text labels for the MDS points.

text.pos**:** Alignment position of the text labels respective to its points (1,2,3,4).

point.col**:** Color for the MDS points / spheres.

text.col**:** Color for the MDS text labels.

point.pch**:** PCH type for the MDS points.

type.3d: Use 'p' for points, 's' for spheres.

radius: Radius for the spheres on a 3D MDS plot. Automatically generated from point.cex and the number of points in the MDS.

app: Appearance of the 3D spheres on a 3D MDS plot, can be 'fill', 'lines', 'grid'.

alpha: Number between 0 and 1 with the level of transparency to be used on spheres on a 3D MDS.

scalecol: Set to TRUE to use a color scale for points, for instance to color points (genes) according to their expression level on a chroGPS-genes MDS plot.

scale: Scale to use to generate scale colors (for instance normalized gene expression for each element (gene) on chroGPS-genes MDS).

palette: Color palette to be used for scale colors.

Objects from the Class

Objects can be created by calls of the form new("clusGPS", ...).

Slots

h: Object of class "hclust" with Agglomerative Nesting or user-provided cluster object.

clus: Object of class "list" with probability contour and bayesian posterior probability of mis-classification information for the clusters evaluated.

adjusted: Object of class "logical" indicating if bayesian posterior probabilities of mis-classification are adjusted for multiple testing.

Author(s)

Oscar Reina

Examples

```
showClass("clusGPS")
```

---

distGPS *Compute matrix with pairwise distances between objects. Several GPS metrics are available.*

Description

The function computes pairwise distances between invididuals (e.g. samples or genes) according to a user-specified metric. Several metrics are available. The precise definition of each metric depends on the class of the first argument (see details section).

Usage

```
distGPS(x, metric='tanimoto', uniqueRows=FALSE, genomelength=NULL, mc.cores=1)
```

Arguments

x	Object for which we want to compute distances
metric	Desired distance metric. Valid options for chroGPS-factors map are 'tanimoto', 'avgdist', 'chisquare' and 'chi' (see details). For chroGPS-genes maps, metrics 'wtanimoto', 'euclidean' and 'manhattan' are also available.
uniqueRows	If set to TRUE and x is a matrix or data.frame, duplicated rows are removed prior to distance calculation. This can save substantial computing time and memory. Notice however that the dimension of the distance matrix is equal to the number of unique rows in x, instead of nrow(x).
genomelength	For 'chi' and 'chisquare' metrics, numeric value indicating the length of the genome. If not given the function uses the minimum length necessary to fit the total length of the result.
mc.cores	If mc.cores>1 and multicore package is loaded, computations are performed in parallel with mc.cores processors when possible.

Details

For RangedDataList objects, distances are defined as follows.

Let a1 and a2 be two RangedData objects. Define as n1 the number of a1 intervals overlapping with some interval in a2. Define n2 analogously. The Tanimoto distance between a1 and a2 is defined as (n1+n2)/(nrow(z1)+nrow(z2)). The average distance between a1 and a2 is defined as .5*(n1/nrow(z1) + n2/nrow(z2)). The wtanimoto distance in chroGPS-genes weights each epigenetic factor (table columns) according to its frequency (table rows). The chi-square distance is defined as the usual chi-square distance on a binary matrix B which is automatically computed by distGPS. The binary matrix B is the matrix with length(x) rows and number of columns equal to the genome length, where B[i,j]==1 indicates that element i has a binding site at base pair j. The chi distance is simply defined as the square root of the chi-square distance. Finally, euclidean and manhattan metrics have the same definition than in the base R function dist.

When choosing a metric one should consider the effect of outliers, i.e. samples with large distance to all other samples. Tanimoto and Average Distance take values between 0 and 1, and therefore outlying distances have a limited effect. Chi-square and Chi distances are not limited between 0 and 1, i.e. some distances may be much larger than others. The Chi metric is slightly more robust to outliers than the Chi-square metric.

For matrix or data.frame objects, x must be a matrix with 0's and 1's (or FALSE and TRUE). The usual definitions are used for Tanimoto (which is equivalent to Jaccard's index), Chi-square and Chi. Average overlap between rows i and j is simply the average between the proportion of elements in i also in j and the proportion of elements in j also in i.

Value

Object of class distGPS, with matrix of pairwise dissimilarities (distances) between objects.

Methods

distGPS:

Each element in x is assumed to indicate the binding sites for a different sample, e.g. epigenetic factor. Typically space(x) indicates the chromosome, start(x) the start position and end(x) the end position (in bp). Strand information is ignored.

signature(x='RangedDataList') **signature(x='matrix')** Rows in x contain individuals for which we want to compute distances. Columns in x contain the variables, and should only contain either 0's and 1's or FALSE and TRUE.

splitDistGPS:

This is a set of internal classes and functions to be used in the parallel computation of Multidimensional Scaling.

uniqueCount:

This function collapses a chroGPS-genes matrix or data frame so that elements with the same combination of variables are aggregated into a single entry. Elements become then identified by their unique pattern and a frequency count is also returned.

See Also

mds to create MDS-oriented objects, procrustesAdj for Procrustes adjustment.

Examples

```
x <- rbind(c(rep(0,15),rep(1,5)),c(rep(0,15),rep(1,5)),c(rep(0,19),1),c(rep(1,5),rep(0,15)))
rownames(x) <- letters[1:4]
d <- distGPS(x,metric='tanimoto')
du <- distGPS(x,metric='tanimoto',uniqueRows=TRUE)
mds1 <- mds(d)
mds1
plot(mds1)
d <- distGPS(x,metric='chisquare')
mds1 <- mds(d)
mds1
plot(mds1)
```

---

distGPS-class *Class* "distGPS"

Description

Pairwise distances between elements. Function distGPS creates objects of this class. splitDistGPS in an private class used internally for parallel Multidimensional Scaling.

Objects from the Class

Objects can be created by calls of the form new("distGPS", ...).

Slots

d**:** Object of class "matrix" with pairwise dissimilarities (distances) between elements.

metric**:** Object of class "character" indicating the metric type used for calculating distances. See function distGPS.

type**:** Object of class "character", deprecated.

Author(s)

Oscar Reina

Examples

```
showClass("distGPS")
```

---

domainDist *Overview of intra and inter-domain distances.*

Description

Given a distance of pairwise distances or dissimilarities between elements, return intra and inter-group sets of distances based on a given group definition. This is useful to get an insight on domain robustness for functional related genes or factors.

Usage

```
domainDist(d, gps='factors', domain, type='intra', col='white', avg=FALSE,
plot=TRUE, ...)
```

Arguments

d	Distance/Dissimilarities matrix, usually the slot d on a distGPS object, but any distance matrix can be given as input.
gps	'factors' for a chroGPS-factors distance matrix, 'genes' for a chroGPS-genes one.
domain	Character vector with group identity for each element d. It can be a functional domain classification (i.e. 'Activation', 'Repression', etc), given for each factor on a chroGPS-factors map or for each gene in a chroGPS-genes map. However, any classification of interest can be used (pathways, gene ontology, etc.)
type	Intradomain ('intra') or Interdomain ('inter') distance overview.
col	Character vector with colors to be passed to plot.
avg	TRUE to return also the average inter or intra domain distance.
plot	TRUE to generate inter/intra domain boxplots.
...	Additional parameters given to the generic function plot.

Value

List of inter or intra domain distances.

Examples

```
# Not run, see further examples in chroGPS-manual.pdf document
# data(sl2)
# data(bg3)
# d1 <- distGPS(sl2,metric='avgdist',mc.cores=8)
# d2 <- distGPS(c(sl2,bg3),metric='avgdist',mc.cores=8)
# domains.sl2 <- as.character(sl2names$Color)
# domains.bg3 <- as.character(bg3names$Color)
# dd.sl2 <- domainDist(d1@d,gps='factors',domain=domains.sl2,type='intra',col=sort(unique(domains.sl2)),avg=TRU
# dd.sl2.bg3 <- domainDist(d2@d,gps='factors',domain=c(domains.sl2,domains.bg3),type='intra',col=sort(unique(dom
```

geneSample *Sample binding site data from Drosophila melanogaster S2-DSRC cell lines (ChIP-chip), coming from modEncode Release 21.*

Description

Sample binding site data from Drosophila melanogaster S2-DSRC cell lines (ChIP-chip), coming from modEncode Release 21. Binding sites are annotated with the Drosophila melanogaster genome version on Biomart ENSEMBL 66 (February 2012) using the annotatePeakInBatch function from the ChIPpeakAnno package. A gap of 1000bp is considered for feature overlap, filtering the resulting features to keep only stricts overlaps within those 1000bp.

Usage

```
data(geneSample)
```

Source

http://www.modencode.org

References

http://www.modencode.org

Examples

```
data(geneSample)
## maybe str(geneSample) ; plot(geneSample) ...
```

geneSetGPS *Highlight point (gene) position over a Multi-dimensional Scaling plot.*

Description

Given a list of genes of interest, the function highlights their position over a Multi-dimensional Scaling plot.

Usage

```
geneSetGPS(x, m, genes, uniqueCount = TRUE, ...)
```

Arguments

x	Matrix or data frame of observations x variables (tipically genes x epigenetic factors), with gene identifiers as rownames.
m	Object of class mds with a valid Multidimensional Scaling representation for the elements in x.
genes	Character vector containing gene identifiers, matching those on rownames(x).
uniqueCount	Set to FALSE if the MDS has been generated directly from the data in x, otherwise set to TRUE to match gene identifiers with their unique pattern of observed variables.
...	Additional parameters given to the generic function plot.

Value

Matrix with coordinates on the given input MDS object for the genes selected.

Author(s)

Oscar Reina

Examples

```
# Not run, see further examples in chroGPS-manual.pdf document
# data(geneSample)
# # Non-unique rows
# d <- distGPS(geneSample,metric='tanimoto',uniqueRows=FALSE)
# mds1 <- mds(d)
# pts <- geneSetGPS(geneSample,mds1,c('FBgn0051048','FBgn0010406','FBgn0034642'),uniqueCount=FALSE)@point
# plot(mds1)
# points(pts,col='red',cex=3)
# # Unique rows
# d <- distGPS(geneSample,metric='tanimoto',uniqueRows=TRUE)
# mds1 <- mds(d)
# plot(mds1)
# pts <- geneSetGPS(geneSample,mds1,c('FBgn0051048','FBgn0010406','FBgn0034642'),uniqueCount=TRUE)@point
# points(pts,col='red',cex=3)
```

---

getURL *Retrieve file from URL.*

Description

A function that can be used to retrieve any file of interest from the internet, in our case, modEncode binding site information GFF files into the working directory. See also help for function gff2RDList.

Usage

```
getURL(urls, filenames, extension='.gff3', method='internal')
```

Arguments

urls	Character vector with one or more target URLs to download.
filenames	Character vector with the filename for each URL target.
extension	If desired, an extension to append to filenames.
method	Either 'internal' to use the system's default or 'wget' if it is installed.

Value

Message indicating the path to downloaded file(s).

Examples

```
# Not run
#getURL('http://www.google.com/index.html','index','.html')
```

gff2RDList *Retrieve binding site information from GFF3 files.*

Description

An auxiliary function to retrieve binding site information from GFF3 format files (for instance those downloaded from modEncode, see function getURL).

Usage

gff2RDList(filenames,listnames,dir,quote=NULL,chrprefix='')

Arguments

filenames	GFF3 filenames to read.
listnames	Names for each read filename, will be used as names of the returned RangedDataList. If not given, filenames will be used as listnames.
dir	Directory where the GFF3 files are located.
quote	Quote character used in the GFF3 files around fields, if any.
chrprefix	Prefix to be placed before the chromosome names if desired, for instance 'chr'.

Value

A list with Enriched and Depleted binding sites, each one is an object of class RangedDataList with the RangedData objects containing the respective enriched or depleted binding sites from each GFF3 file.

Examples

```
# Not run
#getURL('http://intermine.modencode.org/release-30/features.do?type=submission&action=export&format=gff3&su
#test <- gff2RDList('test.gff3',dir=getwd())
#test
#test$Enriched[[1]]
#test$Depleted[[1]]
```

mds *Metric and non-metric Multidimensional Scaling for a distGPS object.*

Description

Generation of Multidimensional Scaling objects for the dissimilarities between elements given as an input in a distGPS object. Metric and non-metric algorithms are available, as well as an optimization algorithm for improving r-square correlation between observed and approximated distances. The MDS calculation for a given distance matrix can be splitted into smaller individual tasks and run in parallel, greatly improving CPU time and system memory usage.

Usage

mds(d, m = NULL, k = 2, type = "classic", add = FALSE, cor.method = "pearson", splitMDS = FALSE, split =

Arguments

d	Object of class distGPS with the pairwise observed dissimilarities between elements, a distance matrix.
m	(Optional). Object of class mds with a MDS object generated from the distances in d. Only MDS type "boostMDS" is available. The mds function performs an optimization of the approximated distances in m in order to improve r-square correlation between them and the observed dissimilarities en d, maximizing goodness of fit.
k	Dimensionality of the reconstructed space, typically set to 2 or 3.
type	Set to "classic" to perform classical MDS (uses function cmdscale from package stats). Set to "isoMDS" to use Kruskal's non-metric MDS (uses function isoMDS from package MASS) Set to "boostMDS" to perform r-square optimization of a pre-computed input MDS for that distance matrix.
add	Logical indicating if an additive constant c* should be computed, and added to the non-diagonal dissimilarities such that all n-1 eigenvalues are non-negative in cmdscale.
cor.method	A character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated.
splitMDS	Set to TRUE to perform computation of the MDS in parallel (see parameters below).
split	Proportion of elements to include in each (but last) distance submatrix.
overlap	Proportion of elements to be used as common anchor points between two adjacent distance submatrixes. These points will be used as spatial references to stitch each two adjacent MDS objects by Procrustes.
reshuffle	Set to TRUE to perform random resampling of the input distance matrix before splitting it for parallel computation. This is often necessary to sufficiently capture the inherent variability of the data in each distance submatrix so that the stitching process can work properly, as the original data may present an arbitrary sorting of some kind. If a previous resampling of the data has been performed, this is not necessary.
set.seed	Random seed to perform the resampling.
mc.cores	Number of cores to be passed to the mclapply function from the multicore package, used to perform the parallel MDS computations.
...	Additional parameters passed to cmdscale, isoMDS or boostMDS, see each individual help file for details.

Value

The function returns a mds object. See help ("mds-Class") for details.

Methods

mds signature(d = "distGPS", m = "missing"): Creates a mds object with points in a k-dimensional space approximating the pairwise distances in d.

mds signature(d = "distGPS", m = "mds"): For the observed dissimilarities in d and a valid spatial representation of them in m, the function returns a mds object with an optimized representation of d in terms of R-square. See help for boostMDS for details.

plot signature(m = "mds"): S4 plot method for mds objects.

Author(s)

Oscar Reina

References

See functions cmdscale, isoMDS from package MASS

Examples

```
x <- rbind(c(rep(0,15),rep(1,5)),c(rep(0,15),rep(1,5)),c(rep(0,19),1),c(rep(1,5),rep(0,15)))
rownames(x) <- letters[1:4]
d <- distGPS(x,metric='tanimoto',uniqueRows=TRUE)
mds1 <- mds(d)
mds1
plot(mds1)
```

---

mds-class *Class* "mds"

Description

Multidimensional Scaling. Function mds creates object of this class.

Details

Parameters for the S4 plot method for mds objects.

Object of class "mds" with a 2D or 3D Multidimensional Scaling to be plotted.

drawlabels: TRUE to use rownames of the MDS points as text labels.

labels: Alternative character vector giving the text labels for the MDS points.

plantar: If a 3D MDS is used, set plantar to TRUE to plot projected views of the MDS using XY, YZ and XZ axis decomposition.

point.cex: Size of the points / spheres for the MDS plot.

text.cex: Size of text labels for the MDS points.

text.pos: Alignment position of the text labels respective to its points (1,2,3,4).

point.col: Color for the MDS points / spheres.

text.col: Color for the MDS text labels.

point.pch: PCH type for the MDS points.

type.3d: Use 'p' for points, 's' for spheres.

radius: Radius for the spheres on a 3D MDS plot. Automatically generated from point.cex and the number of points in the MDS.

app: Appearance of the 3D spheres on a 3D MDS plot, can be 'fill', 'lines', 'grid'.

alpha: Number between 0 and 1 with the level of transparency to be used on spheres on a 3D MDS.

scalecol: Set to TRUE to use a color scale for points, for instance to color points (genes) according to their expression level on a chroGPS-genes MDS plot.

scale: Scale to use to generate scale colors (for instance normalized gene expression for each element (gene) on chroGPS-genes MDS).

palette: Color palette to be used for scale colors.

Objects from the Class

Objects can be created by calls of the form new("mds", ...).

Slots

points: Object of class "matrix" with coordinates in the approximated space.

R.square: Object of class "numeric" with the percentage of variability from the original dissimilarities captured in the approximated distances.

Author(s)

Oscar Reina

See Also

cmdscale from package base. isoMDS from package MASS.

Examples

showClass("mds")

---

procrustesAdj *Use Procrustes to adjust an MDS map containing samples obtained under different conditions, e.g. technology or genetic backgrounds.*

Description

The function adjusts a previous mds to take into account that samples were obtained under different conditions, e.g. technological or genetic. Pairwise adjustments are performed by identifying samples present in both conditions and using Procrustes. When there are more than two conditions, sequential pairwise adjustments are applied (in the order that maximizes the number of common samples in each pairwise adjustment).

Usage

procrustesAdj(mds1, d, adjust, sampleid)

Arguments

mds1	Object of class mds with a Multi-dimensional scaling analysis on a distance matrix, typically obtained by a previous call to mds.
d	Object of class distGPS with the matrix used to create the Multidimensional Scaling object usually through a call to mds.
adjust	Vector indicating the adjustment factor, i.e. the condition under which each sample has been obtained.
sampleid	Vector containing the sample identifier. sampleid should take the same value for samples obtained under different conditions, as this is used to detect the samples to be used for Procrustes adjustment.

Details

We implement the Procrustes adjustment as follows. First we identify common samples, i.e. those obtained both under conditions A and B. Second, we use Procrustes to estimate the shift, scale and rotation that best matches the position of the samples in B to those in A. If only 1 sample was obtained under both conditions, only the shift is estimated. Last, we apply the estimated shift, scale and rotation to all B samples. That is, the Procruses parameters are estimated using common samples only, which are then applied to all samples to perform the adjustment.

Notice that the R square of the adjusted mds is typically improved after Procrustes adjustment, since distances between samples obtained under different conditions are set to NA and therefore MDS needs to approximate distances between less points.

When several replicates are available for a given sampleid under the same condition (adjust), the average position of all replicates is used.

Value

Adjusted mds object. Have in mind that only original distances between samples obtained under the same condition should be conserved, as the adjusted distances manipulated by Procrustes no longer correlate with the distances between their points in the adjusted MDS.

Methods

signature(x='mds') x is a mds object with the results of an MDS analysis.

See Also

distGPS for computing distances, mds to create MDS-oriented objects.

Examples

```
st1 <- runif(100,1,1000); st2 <- runif(100,500,1500)  #Peak starts
st3 <- runif(100,1000,2000); st4 <- runif(100,1500,2000)
#cond1: more precise technology
cond1 <- RangedDataList(s1=RangedData(IRanges(st1,st1+100)),s2=RangedData(IRanges(st2,st2+100)),s3=Ranged
#cond2: less precise
cond2 <- RangedDataList(s1=RangedData(IRanges(st1-200,st1+300)),s2=RangedData(IRanges(st2-200,st2+300)),s5=
x <- c(cond1,cond2)
d <- distGPS(x,metric='tanimoto')  #compute distances
mds1 <- mds(d)  #MDS
#Adjust via Procrustes
mds2 <- procrustesAdj(mds1,d,adjust=rep(c('seq','chip'),each=3),sampleid=names(x))
plot(mds1)
plot(mds2)
#Adjust via peak width
xadj <- adjustPeaks(x,adjust=rep(c('seq','chip'),each=3),sampleid=names(x))
dadj <- distGPS(xadj)
mds3 <- mds(dadj)
plot(mds3)
```

sl2 *Sample binding site and related data from Drosophila melanogaster.*

Description

chroGPS example ChIP-CHIP data for S2 and BG3 cell lines (modEncode), ChIP-Seq data for S2 cell lines (NCBI GEO GSE19325).

Usage

```
data(sl2)
```

Source

http://www.modencode.org http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE19325

References

http://www.modencode.org http://www.ncbi.nlm.nih.gov/geo/

Examples

```
data(sl2)
```

splitDistGPS-class *Class* "splitDistGPS"

Description

Set of pairwise distances between elements. This is an internal class to be used with the parallel version of mds, and should not be used on its own.

Objects from the Class

Objects from this class are used internally for parallel Multidimensional Scaling. See mds for details.

Slots

d: List of distGPS objects.

size: Object of class "numeric" indicating the size of the individual distGPS objects in the list. See function mds.

o: Object of class "numeric" with the overlap (anchor points) between adjacent distGPS objects. See function mds.

shuffle: Object of class "numeric", deprecated.

Author(s)

Oscar Reina

Examples

```
showClass("splitDistGPS")
```

Index