Package 'iClusterPlus'

November 12, 2026

Title Integrative clustering of multi-type genomic data

Version 1.48.0

Date 2026-03-20

Depends R (>= 4.1.0), irlba, parallel

Suggests RUnit, BiocGenerics

Author Qianxing Mo, Ronglai Shen

Maintainer Qianxing Mo <qianxing.mo@moffitt.org>, Ronglai Shen <shenr@mskcc.org>

Description Integrative clustering of multiple genomic data using a joint latent variable model.

LazyData yes

LazyDataCompression bzip2

License GPL (>= 2)

biocViews Multi-omics, Clustering

git_url https://git.bioconductor.org/packages/iClusterPlus

git_branch RELEASE_3_23

git_last_commit 5b2bd1a

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

breast.chr17 2
CNregions 3
compute.pod 4
coord 5
gbm 5
glp 6
iCluster 6
iCluster2 8

iCluster2b 10
iClusterBayes 13
iClusterBayes2 14
iClusterPlus 16
iClusterPlus2 18
pcaVarPlot 19
plotHeatmap 20
plotHMBayes 22
plotiCluster 23
plotRI 24
tune.iCluster2 25
tune.iCluster2b 26
tune.iClusterBayes 28
tune.iClusterPlus 30
UM 31
utility 32
variation.hg18.v10.nov.2010 34

Index **35**

`breast.chr17` *Breast cancer data set DNA copy number and mRNA expression measure on chromosome 17*

Description

This is a subset of the breast cancer data from Pollack et al. (2002).

Usage

```
data(breast.chr17)
```

Format

A list object containing two data matrices: DNA and mRNA. They consist chromosome 17 data in 41 samples (4 cell lines and 37 primary tumors).

Source

This data can be downloaded at http://www.pnas.org/content/99/20/12963/suppl/DC1

References

Pollack, J.R. et al. (2002) Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. Proc. Natl Acad. Sci. USA, 99, 12963-12968.

CNregions *A function to remove redundant copy number regions*

Description

This function is used to reduce copy number regions.

Usage

```
CNregions(seg, epsilon=0.005, adaptive=FALSE, rmCNV=FALSE, cnv=NULL,
        frac.overlap=0.5, rmSmallseg=TRUE, nProbes=5)
```

Arguments

seg	DNAcopy CBS segmentation output.
epsilon	the maximum Euclidean distance between adjacent probes tolerated for denying a nonredundant region. epsilon=0 is equivalent to taking the union of all unique break points across the n samples.
adaptive	Vector of length-m lasso penalty terms.
rmCNV	If TRUE, remove germline CNV.
cnv	A data frame containing germline CNV data.
frac.overlap	Fraction of overlap between 2 segments. If rmCNV=TRUE, overlapped segments will be removed if the overlapped fraction >= fra.overlap.
rmSmallseg	If TRUE, remove small segment.
nProbes	The segment length threshold below which the segment will be removed if rmSmallseq = TRUE.

Value

A matrix with reduced copy number regions.

Author(s)

Ronglai Shen <shenr@mskcc.org>

References

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. (2013). Pattern discovery and cancer gene identification in integrated cancer genomic data. Proc. Natl. Acad. Sci. USA.

See Also

breast.chr17,plotiCluster, compute.pod,iCluster,iClusterPlus

Examples

```
#data(gbm)
#library(GenomicRanges)
#library(cluster)
#reducedM=CNregions(seg,epsilon=0,adaptive=FALSE,rmCNV=TRUE,cnv=NULL,
#  frac.overlap=0.5, rmSmallseg=TRUE,nProbes=5)
```

`compute.pod` *A function to compute the proportion of deviation from perfect block diagonal matrix*

Description

A function to compute the proportion of deviation from perfect block diagonal matrix.

Usage

```
compute.pod(fit)
```

Arguments

`fit`	A iCluster object

Value

pod	proportion of deviation from perfect block diagonal matrix

Author(s)

Ronglai Shen <shenr@mskcc.org>

References

Ronglai Shen, Adam Olshen, Marc Ladanyi. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906-2912.

See Also

`iCluster`,`iCluster2`, `plotiCluster`

Examples

```
# library(iCluster)
# data(breast.chr17)
# fit=iCluster(breast.chr17, k=4, lambda=c(0.2,0.2))
# plotiCluster(fit=fit, label=rownames(breast.chr17[[2]]))
# compute.pod(fit)
```

coord *genomic coordinates*

Description

genomic coordinates for the copy number data in gbm

Usage

```
data(coord)
```

Format

A data matrix consists of chr number, start and end position for the genes included in the gbm copy number data.

References

Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E. Seshan, Adam B. Olshen, Jason Huse, Marc Ladanyi, Chris Sander. (2012). Integrative Subtype Discovery in Glioblastoma Using iCluster. *PLoS ONE* 7, e35236

gbm *GBM data*

Description

This is a subset of the glioblastoma dataset from the cancer genome atlas (TCGA) GBM study (2009) used in Shen et al. (2012).

Usage

```
data(gbm)
```

Format

A list object containing three data matrices: copy number, methylation and mRNA expression in 84 samples.

Value

`gbm.seg`	GBM copy number segmentation results genereated by DNAcopy package.
`gbm.exp`	GBM gene expression data.
`gbm.mut`	GBM mutation data.

References

Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E. Seshan, Adam B. Olshen, Jason Huse, Marc Ladanyi, Chris Sander. (2012). Integrative Subtype Discovery in Glioblastoma Using iCluster. *PLoS ONE* 7, e35236

`glp` *good lattice points using the uniform design*

Description

good lattice points using the uniform design (Fang and Wang 1995)

Usage

```
data(glp)
```

Format

A list object containing sampling design for s=2-5 where s is the number of tuning parameters.

References

Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E. Seshan, Adam B. Olshen, Jason Huse, Marc Ladanyi, Chris Sander. (2012). Integrative Subtype Discovery in Glioblastoma Using iCluster. *PLoS ONE* 7, e35236

Fang K, Wang Y (1994) Number theoretic methods in statistics. London, UK: Chapman abd Hall.

`iCluster` *Integrative clustering of multiple genomic data types*

Description

Given multiple genomic data types (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iCluster fits a regularized latent variable model based clustering that generates an integrated cluster assigment based on joint inference across data types

Usage

```
iCluster(datasets, k, lambda, scalar=FALSE, max.iter=50,epsilon=1e-3)
```

Arguments

datasets	A list object containing m data matrices representing m different genomic data types measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
k	Number of subtypes.
lambda	Vector of length-m lasso penalty terms.
scalar	If TRUE, assumes scalar covariance matrix Psi. Default is FALSE.
max.iter	Maximum iteration for the EM algorithm.
epsilon	EM algorithm convegence criterion.

Value

A list with the following elements.

meanZ	Relaxed cluster indicator matrix.
beta	Coefficient matrix.
clusters	Cluster assigment.
conv.rate	Convergence history.

Author(s)

Ronglai Shen <shenr@mskcc.org>

References

Ronglai Shen, Adam Olshen, Marc Ladanyi. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906-2912.

See Also

breast.chr17,plotiCluster, compute.pod

Examples

```
data(breast.chr17)
fit=iCluster(breast.chr17, k=4, lambda=c(0.2,0.2))
plotiCluster(fit=fit, label=rownames(breast.chr17[[2]]))
compute.pod(fit)

#library(gplots)
#library(lattice)
#col.scheme = alist()
#col.scheme[[1]] = bluered(256)
#col.scheme[[2]] = greenred(256)
#cn.image=breast.chr17[[2]]
#cn.image[cn.image>1.5]=1.5
#cn.image[cn.image< -1.5]= -1.5
```

```
#exp.image=breast.chr17[[1]]
#exp.image[exp.image>3]=3
#exp.image[exp.image< -3]=3
#plotHeatmap(fit, datasets=list(cn.image,exp.image), type=c("gaussian","gaussian"),
#  row.order=c(FALSE,FALSE), width=5, col.scheme=col.scheme)
```

`iCluster2` *Integrative clustering of multiple genomic data types*

Description

Given multiple genomic data types (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iCluster fits a regularized latent variable model based clustering that generates an integrated cluster assigment based on joint inference across data types

Usage

```
iCluster2(x, K, lambda, method=c("lasso","enet","flasso","glasso","gflasso"),
  chr=NULL, maxiter=50, eps=1e-4, eps2=1e-8)
```

Arguments

`x`	A list object containing m data matrices representing m different genomic data types measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
`K`	Number of subtypes.
`lambda`	A list with m elements; each element is a vector with one or two elements depending on the methods used.
`method`	Method used for clustering and variable selection.
`chr`	Chromosome labels
`maxiter`	Maximum iteration for the EM algorithm.
`eps`	EM algorithm convegence criterion 1.
`eps2`	EM algorithm convegence criterion 2.

Value

A list with the following elements.

`cluster`	Cluster assigment.
`centers`	cluster centers.
`Phivec`	parameter phi; a vector.
`beta`	parameter B; a matrix.
`meanZ`	meanZ
`EZZt`	EZZt
`dif`	difference
`iter`	iteration

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>,Ronglai Shen,Sijian Wang

References

Ronglai Shen, Sijian Wang, Qianxing Mo. (2013). Sparse Integrative Clustering of Multiple Omics Data Sets. *Annals of Applied Statistics.* 7(1):269-294

See Also

plotiCluster, compute.pod, iClusterPlus

Examples

```
## clustering
n1 = 20
n2 = 20
n3 = 20
n = n1+n2+n3
p = 5
q = 100

x = NULL
x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[1]] = cbind(xa,xb)

x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[2]] = cbind(xa,xb)

x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[3]] = cbind(xa,xb)


x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[4]] = cbind(xa,xb)
```

```
x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[5]] = cbind(xa,xb)

method = c('lasso', 'enet', 'flasso', 'glasso', 'gflasso')
lambda=alist()
lambda[[1]] = 30
lambda[[2]] = c(20,1)
lambda[[3]] = c(20,20)
lambda[[4]] = 30
lambda[[5]] = c(30,20)

chr=c(rep(1,10),rep(2,(p+q)-10))
date()
fit2 = iCluster2(x, K=3, lambda, method=method, chr=chr, maxiter=20,eps=1e-4, eps2=1e-8)
date()

par(mfrow=c(5,1),mar=c(4,4,1,1))
for(i in 1:5){
barplot(fit2$beta[[i]][,1])
}

#library(gplots)
#library(lattice)

#plotHeatmap(fit2, datasets=x, type=rep("gaussian",length(x)),
   #row.order=c(TRUE,TRUE,FALSE,TRUE,FALSE),
   #sparse=rep(FALSE,length(x)), scale=rep("row",5), width=5,
   #col.scheme=rep(list(bluered(256)),length(x)))
```

`iCluster2b` *Integrative clustering of multi-omics data*

Description

Given multi-omics data (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iCluster2b fits a regularized latent factor(variable) model that generates an latent factor matrix that can be used for integrative clustering of samples. In addition, the driver features of sample clustering can be identified by sparse coefficient matrices.

Usage

```
iCluster2b(xList,K=3, lambda, method=c("lasso","enet","flasso","glasso","gflasso"),
  chr=NULL, EM.iter=25, eps=1e-4, eps2=1e-6)
```

Arguments

xList	A list object containing m data matrices representing m multi-omics data measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
K	An integer representing the number of the latent factors used for modeling.
lambda	A list with m elements corresponding to the method argument. For lasso and glasso, each element is a positive value. For enet, flasso and gflasso, each element is a vector with two positive values.
method	Method used for variable selection.
chr	Chromosome labels
EM.iter	Maximum iteration for the EM algorithm.
eps	EM algorithm convergence criterion 1.
eps2	EM algorithm convergence criterion 2.

Value

A list with the following elements.

meanZ	A n x k matrix; the rows represent samples and the columns represent the K factors.
beta	A list object containing m coefficient matrices corresponding to the m data matrices.
iter	EM iteration.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>,Ronglai Shen,Sijian Wang

References

Ronglai Shen, Sijian Wang, Qianxing Mo. (2013). Sparse Integrative Clustering of Multiple Omics Data Sets. *Annals of Applied Statistics.* 7(1):269-294

See Also

tune.iCluster2b, iClusterPlus2, iClusterBayes

Examples

```
## clustering
n1 = 20
n2 = 20
n3 = 20
n = n1+n2+n3
p = 5
q = 100
```

```
x = NULL
x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[1]] = cbind(xa,xb)

x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[2]] = cbind(xa,xb)

x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[3]] = cbind(xa,xb)


x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[4]] = cbind(xa,xb)

x1a = matrix(rnorm(n1*p), ncol=p)
x2a = matrix(rnorm(n1*p, -1.5,1), ncol=p)
x3a = matrix(rnorm(n1*p, 1.5, 1), ncol=p)
xa = rbind(x1a,x2a,x3a)
xb = matrix(rnorm(n*q), ncol=q)
x[[5]] = cbind(xa,xb)

method = c('lasso', 'enet', 'flasso', 'glasso', 'gflasso')
lambda=alist()
lambda[[1]] = 30
lambda[[2]] = c(20,1)
lambda[[3]] = c(20,20)
lambda[[4]] = 30
lambda[[5]] = c(30,20)

chr=c(rep(1,10),rep(2,(p+q)-10))
date()
fit2 = iCluster2b(x, K=3, lambda, method=method, chr=chr, EM.iter=20,eps=1e-4, eps2=1e-6)
date()

par(mfrow=c(5,1),mar=c(4,4,1,1))
for(i in 1:5){
barplot(fit2$beta[[i]][,1])
```

```
}
```

`iClusterBayes` *Integrative clustering of multiple genomic data types*

Description

Given multiple genomic data types (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iClusterBayes fits a Bayesian latent variable model that generates an integrated cluster assignment based on joint inference across data types and identifies genomic features that contribute to the clusters.

Usage

```
iClusterBayes(dt1,dt2=NULL,dt3=NULL,dt4=NULL,dt5=NULL,dt6=NULL,
 type = c("gaussian","binomial","poisson"),K=2,n.burnin=1000,n.draw=1200,
 prior.gamma=rep(0.1,6),sdev=0.5,beta.var.scale=1,thin=1,pp.cutoff=0.5)
```

Arguments

`dt1`	Data set 1 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt2`	Data set 2 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt3`	Data set 3 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt4`	Data set 4 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt5`	Data set 5 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt6`	Data set 6 - a matrix with rows and columns representing samples and genomic features, respectively.
`type`	Data type corresponding to dt1-6, which can be gaussian, binomial, or poisson.
`K`	The number of eigen features. Given K, the number of cluster is K+1.
`n.burnin`	Number of MCMC burnin.
`n.draw`	Number of MCMC draw.
`prior.gamma`	Prior probability for the indicator variable gamma of each data set.
`sdev`	Standard deviation of random walk proposal for the latent variable.
`beta.var.scale`	A positive value to control the scale of covariance matrix of the proposed beta.
`thin`	A parameter to thin the MCMC chain in order to reduce autocorrelation. Discard all but every 'thin'th sampling values. When thin=1, all sampling values are kept.
`pp.cutoff`	Posterior probability cutoff for the indicator variable gamma. The BIC and deviance ratio will be calculated by setting parameter beta to zero when the posterior probability of gamma <= cutoff.

Value

A list with the following elements.

alpha	Intercept parameter.
beta	Information parameter.
beta.pp	Posterior probability of beta. The higher the beta.pp, the more likely the beta should be included in the model.
gamma.ar	Acceptance ratio for the parameter gamma.
beta.ar	Acceptance ratio for the parameter beta.
Z.ar	Acceptance ratio for the latent variable.
clusters	Cluster assignment.
centers	Cluster center.
meanZ	The latent variable.
BIC	Bayesian information criterion.
dev.ratio	see dev.ratio defined in glmnet package.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics 19(1):71-86.

See Also

tune.iClusterBayes,plotHMBayes,iClusterPlus,tune.iClusterPlus,plotHeatmap

Examples

```
# see iManual.pdf
```

---

iClusterBayes2 *Integrative clustering of multi-omics data*

Description

Given multi-omics data (e.g., Somatic mutation, copy number, gene expression, DNA methylation) measured in the same set of samples, iClusterBayes2 fits a Bayesian latent factor model that generates an latent factor matrix that can be used for integrative clustering of samples. In addition, the driver features of sample clustering can be identified by the posterior probability of the model parameters.

Usage

```
iClusterBayes2(xList,type = c("gaussian","binomial","poisson"),K=3,n.burnin=1000,n.draw=1200,
 prior.gamma=rep(0.1,6),sdev=0.5,beta.var.scale=1,thin=1,pp.cutoff=0.5)
```

Arguments

xList	A list object containing m data matrices representing m multi-omics data measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features. The allowed maximum number of data matrices is 6 (m < 7).
type	Data type corresponding to the data matrices in xList, which can be gaussian, binomial, or poisson.
K	An integer representing the number of the latent factors used for modeling.
n.burnin	Number of MCMC burnin.
n.draw	Number of MCMC draw.
prior.gamma	Prior probability for the indicator variable gamma of each data set.
sdev	Standard deviation of random walk proposal for the latent variable.
beta.var.scale	A positive value to control the scale of covariance matrix of the proposed beta.
thin	A parameter to thin the MCMC chain in order to reduce autocorrelation. Discard all but every 'thin'th sampling values. When thin=1, all sampling values are kept.
pp.cutoff	Posterior probability cutoff for the indicator variable gamma. The BIC and deviance ratio will be calculated by setting parameter beta to zero when the posterior probability of gamma <= cutoff.

Value

A list with the following elements.

alpha	A list of object for the intercept parameters corresponding to the m data matrices.
beta	A list object containing m coefficient matrices corresponding to the m data matrices.
meanZ	A n x k matrix; the rows represent samples and the columns represent the K factors.
beta.pp	A list of posterior probability for the parameter beta. The higher the beta.pp, the more likely the beta should be included in the model.
gamma.ar	A list of acceptance ratio for the parameter gamma.
beta.ar	A list of acceptance ratio for the parameter beta.
Z.ar	Acceptance ratio for the latent variable.
BIC	Bayesian information criterion.
dev.ratio	see dev.ratio defined in glmnet package.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics 19(1):71-86.

See Also

iClusterPlus2,plotHMBayes,plotHeatmap

Examples

```
# see iManual.pdf
```

`iClusterPlus` *Integrative clustering of multiple genomic data types*

Description

Given multiple genomic data types (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iClusterPlus fits a regularized latent variable model based clustering that generates an integrated cluster assignment based on joint inference across data types

Usage

```
iClusterPlus(dt1,dt2=NULL,dt3=NULL,dt4=NULL,
 type=c("gaussian","binomial","poisson","multinomial"),
  K=2,alpha=c(1,1,1,1),lambda=c(0.03,0.03,0.03,0.03),
 n.burnin=100,n.draw=200,maxiter=20,sdev=0.05,eps=1.0e-4)
```

Arguments

`dt1`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt2`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt3`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt4`	A data matrix. The rows represent samples, and the columns represent genomic features.
`type`	Data type, which can be gaussian, binomial, poisson, multinomial.
`K`	The number of eigen features. Given K, the number of cluster is K+1.
`alpha`	Vector of elasticnet penalty terms. At this version of iClusterPlus, elasticnet is not used. Therefore, all the elements of alpha are set to 1.
`lambda`	Vector of lasso penalty terms.
`n.burnin`	Number of MCMC burnin.

n.draw	Number of MCMC draw.
maxiter	Maximum iteration for the EM algorithm.
sdev	standard deviation of random walk proposal.
eps	Algorithm convergence criterion.

Value

A list with the following elements.

alpha	Intercept parameter.
beta	Information parameter.
clusters	Cluster assignment.
centers	Cluster center.
meanZ	Latent variable.
BIC	Bayesian information criterion.
dev.ratio	see dev.ratio defined in glmnet package.
dif	absolute difference for the parameters in the last and next-to-last iterations.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>,Ronglai Shen, Sijian Wang

References

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. (2013). Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl. Acad. Sci. USA*. 110(11):4245-50.

See Also

plotiCluster,iCluster, compute.pod

Examples

```
# see iManual.pdf
```

`iClusterPlus2` *Integrative clustering of multi-omics data*

Description

Given multi-omics data (e.g., somatic mutation,copy number, gene expression, DNA methylation) measured in the same set of samples, iClusterPlus2 fits a regularized latent factor(variable) model that generates an latent factor matrix that can be used for integrative clustering of samples. In addition, the driver features of sample clustering can be identified by sparse coefficient matrices.

Usage

```
iClusterPlus2(xList,type=c("gaussian","binomial","poisson","multinomial"),K=3,
                  n.burnin=100,n.draw=200,maxiter=25,sdev=0.05,lambda.scale=1/3,
                     BICrate.cutoff=rep(0.01,4),min.shrinkage.rate=rep(0.05,4))
```

Arguments

`xList`	A list object containing m data matrices representing m multi-omics data measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
`type`	Data type, which can be gaussian, binomial, poisson, multinomial.
`K`	An integer representing the number of the latent factors used for modeling.
`n.burnin`	Number of MCMC burnin.
`n.draw`	Number of MCMC draw.
`maxiter`	Maximum iteration for the EM algorithm.
`sdev`	standard deviation of random walk proposal.
`lambda.scale`	scaling factor for the lasso regularization parameter when the data type is binomial.Empirically, lambda.scale is within (0.1, 1).
`BICrate.cutoff`	BIC rate between iteration i+1 and i. If BIC rate < BICrate.cutoff, the search for optimal lambda will stop.
`min.shrinkage.rate`	The minimum lasso shrinkage rates for multi-omics features.

Value

A list with the following elements.

`alpha`	A list of object for the intercept parameters corresponding to the m data matrices.
`beta`	A list object containing m coefficient matrices corresponding to the m data matrices.
`meanZ`	A n x k matrix; the rows represent samples and the columns represent the K factors.
`BIC`	Bayesian information criterion.
`dev.ratio`	see dev.ratio defined in glmnet package.
`lambda`	Final lasso regularization parameters used for iCluster modeling.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. (2013). Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl. Acad. Sci. USA*. 110(11):4245-50.

See Also

tune.iCluster2b, iClusterPlus, iClusterBayes

Examples

```
# see iManual.pdf
```

---

pcaVarPlot *A function to make PCA variance plot of the combined matrix of multi-omics data matrices.*

Description

Multi-omics data matrices are column-combined and then the PCA variance plot of the combined matrix is made.

Usage

```
pcaVarPlot(xList,K=10)
```

Arguments

xList	A list object containing m data matrices representing m multi-omics data measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
K	The number of principle components.

Value

no value returned.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

TBD

See Also

tune.iCluster2b, iClusterPlus2, iClusterBayes, plotHeatmap

Examples

```
# see tutorials
```

---

plotHeatmap *A function to generate heatmap panels sorted by integrated cluster assignment.*

Description

A function to generate heatmap panels sorted by integrated cluster assignment.

Usage

```
plotHeatmap(fit, xList, type=c("gaussian","binomial","poisson","multinomial"), sample.order=NULL, fea
        dist.method = "euclidean", hclust.method="ward.D",sparse=NULL, threshold=rep(0.25,length(xLis
        feature.scale=rep(F,length(xList)), col.scheme=rep(list(bluered(256)),length(xList)), width=
            chr=NULL, plot.chr=NULL, cap=rep(0,length(xList)))
```

Arguments

fit	A iCluster object.
xList	A list object of data matrices.
type	Types of data in the xList.
sample.order	User supplied cluster assignment.
feature.order	A vector of logical values each specify whether the genomic features in the corresponding data matrix should be reordered by similarity. Default is TRUE.
dist.method	Method used to calculate distance (similarity) between features. Default method is "euclidean". Another choice is "correlation", which is Pearson correlation coefficient.
hclust.method	Method passed to hclust function. See hclust for details.
sparse	A vector of logical values each specify whether to plot the top cluster-discriminant features. Default is FALSE.
threshold	When sparse is TRUE, a vector of threshold values to include the genomic features for which the absolute value of the associated coefficient estimates fall in the top quantile. threshold=c(0.25,0.25) takes the top quartile most discriminant features in data type 1 and data type 2 for plot.

feature.scale	A vector of logical values each specify whether data should be scaled. Default is FALSE.
col.scheme	Color scheme. Can use bluered(n) in gplots R package.
width	Width of the figure in inches.
chr	A vector of chromosome number.
plot.chr	A vector of logical values each specify whether to annotate chromosome number on the left of the panel. Typically used for copy number data type. Default is FALSE.
cap	A numeric vector used to control the heatmap colors. For example, cap=c(0,0.0.95,0.95) indicates that no truncation for the image data used to make heatmap 1 data, and the data used to make heatmaps 2 and 3 are truncated at 95% quantile.

Details

The samples are ordered by the cluster assignment using the R code: order(fit$clusters). For each data set, the features are ordered by hierarchical clustering of the features using the hclust.method and euclidean (or 1-correlation coefficient) as the distance.

Value

feature.hclust	A list of objects returned by "hclust" that describes the tree produced by the clustering process. For a given data matrix, if feature.order is TRUE, the features of the data matrix are ordered by the tree generated by "hclust". Please see "hclust" for details.
image.data	A list of data matrices used to make the heatmaps.

Author(s)

Ronglai Shen <shenr@mskcc.org>; Qianxing Mo <qianxing.mo@moffitt.org>

References

Ronglai Shen, Adam Olshen, Marc Ladanyi. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906-2912.

Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E. Seshan, Adam B. Olshen, Jason Huse, Marc Ladanyi, Chris Sander. (2012). Integrative Subtype Discovery in Glioblastoma Using iCluster. *PLoS ONE* 7, e35236

See Also

iCluster,iCluster2b, iClusterPlus2

Examples

```
# see iManual.pdf
```

plotHMBayes *A function to generate heatmap panels sorted by integrated cluster assignment.*

Description

A function to generate heatmap panels sorted by integrated cluster assignment.

Usage

```
plotHMBayes(fit, xList, type = c("gaussian", "binomial", "poisson"),
   sample.order = NULL, feature.order = NULL, dist.method="euclidean", hclust.method="ward.D",
   sparse = NULL, threshold = rep(0.5,length(xList)), feature.scale = rep(F,length(xList)),
    col.scheme = rep(list(bluered(256)),length(xList)), width=5, chr=NULL,
    plot.chr=NULL, cap=rep(0,length(xList)))
```

Arguments

fit	A iClusterBayes object.
xList	A list object of data matrices.
type	Types of data in the xList.
sample.order	User supplied cluster assignment.
feature.order	A vector of logical values each specify whether the genomic features in the corresponding data matrix should be reordered by similarity. Default is TRUE.
dist.method	Method used to calculate distance (similarity) between features. Default method is "euclidean". Another choice is "correlation", which is Pearson correlation coefficient.
hclust.method	Method passed to hclust function. See hclust for details.
sparse	A vector of logical values each specify whether to plot the top cluster-discriminant features. Default is FALSE.
threshold	When sparse is TRUE, a vector of threshold values to include the genomic features on the heatmap. Each data set should have a threshold. For each data set, a feature with posterior probability greater than the threshold will be included. Default value is 0.5 for each data set.
feature.scale	A vector of logical values each specify whether data should be scaled. Default is FALSE.
col.scheme	Color scheme. Can use bluered(n) in gplots R package.
width	Width of the figure in inches.
chr	A vector of chromosome number.
plot.chr	A vector of logical values each specify whether to annotate chromosome number on the left of the panel. Typically used for copy number data type. Default is FALSE.
cap	A numeric vector used to control the heatmap colors. For example, cap=c(0,0.0.95,0.95) indicates that no truncation for the image data used to make heatmap 1 data, and the data used to make heatmaps 2 and 3 are truncated at 95% quantile.

Details

The samples are ordered by the cluster assignment using the R code: order(fit$clusters). For each data set, the features are ordered by hierarchical clustering of the features using the hclust.method and euclidean (or 1-correlation coefficient) as the distance.

Value

`feature.hclust` A list of objects returned by "hclust" that describes the tree produced by the clustering process. For a given data matrix, if feature.order is TRUE, the features of the data matrix are ordered by the tree generated by "hclust". Please see "hclust" for details.

`image.data` A list of data matrices used to make the heatmaps.

no value returned.

Author(s)

Ronglai Shen <shenr@mskcc.org>,Qianxing Mo <qianxing.mo@moffitt.org>

References

Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics 19(1):71-86.

See Also

iClusterBayes,plotHeatmap

Examples

```
# see iManual.pdf
```

---

plotiCluster *A function to generate cluster separability matrix plot.*

Description

A function to generate cluster separability matrix plot.

Usage

```
plotiCluster(fit,label=NULL)
```

Arguments

`fit`	A iCluster object
`label`	Sample labels

Value

no value returned.

Author(s)

Ronglai Shen <shenr@mskcc.org>

References

Ronglai Shen, Adam Olshen, Marc Ladanyi. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906-2912.

See Also

iCluster, compute.pod

Examples

```
# library(iCluster)
# data(breast.chr17)
# fit=iCluster(datasets=breast.chr17, k=4, lambda=c(0.2,0.2))
# plotiCluster(fit=fit, label=rownames(breast.chr17[[2]]))
# compute.pod(fit)
```

---

plotRI *A function to generate reproducibility index plot.*

Description

A function to generate reproducibility index plot.

Usage

```
plotRI(cv.fit)
```

Arguments

cv.fit	A tune.iCluster2 object

Value

no value returned.

Author(s)

Ronglai Shen <shenr@mskcc.org>

References

Ronglai Shen, Adam Olshen, Marc Ladanyi. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906-2912.

Ronglai Shen, Qianxing Mo, Nikolaus Schultz, Venkatraman E. Seshan, Adam B. Olshen, Jason Huse, Marc Ladanyi, Chris Sander. (2012). Integrative Subtype Discovery in Glioblastoma Using iCluster. *PLoS ONE* 7, e35236

See Also

iCluster

Examples

```
#data(simu.datasets)
#cv.fit=alist()
#for(k in 2:5){
#  cat(paste("K=",k,sep=""),'\n')
#  cv.fit[[k]]=tune.iCluster2(datasets=simu.datasets, k,nrep=2, n.lambda=8)
#}

##Reproducibility index (RI) plot
#plotRI(cv.fit)
```

`tune.iCluster2` *Integrative clustering of multiple genomic data types*

Description

Given multiple genomic data types (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, iCluster fits a regularized latent variable model based clustering that generates an integrated cluster assignment based on joint inference across data types

Usage

```
tune.iCluster2(x, K, method=c("lasso","enet","flasso","glasso","gflasso"),base=200,
 chr=NULL,true.class=NULL,lambda=NULL,n.lambda=NULL,save.nonsparse=F,nrep=10,eps=1e-4)
```

Arguments

`x`	A list object containing m data matrices representing m different genomic data types measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
`K`	Number of subtypes.
`lambda`	User supplied matrix of lambda to tune.
`method`	Method used for clustering and variable selection.

chr	Chromosome labels
n.lambda	Number of lambda to sample using uniform design.
nrep	Fold of cross-validation.
base	Base.
true.class	True class label if available.
save.nonsparse	Logic argument whether to save the nonsparse fit.
eps	EM algorithm convergence criterion

Value

A list with the following elements.

best.fit	Best fit.
best.lambda	Best lambda.
ps	Rand index
ps.adjusted	Adjusted Rand index.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>,Ronglai Shen,Sijian Wang

References

Ronglai Shen, Sijian Wang, Qianxing Mo. (2013). Sparse Integrative Clustering of Multiple Omics Data Sets. *Annals of Applied Statistics.* 7(1):269-294

See Also

iCluster2

---

tune.iCluster2b *Integrative clustering of multi-omics data*

Description

This function finds optimal lasso regularization parameters for iCluster2b.

Usage

```
tune.iCluster2b(xList,K=3,method=c("lasso","enet"),min.lambda=10,max.lambda=500,
  lambda.iter=25,EM.iter=25,min.shrinkage.rate=rep(0.05,length(xList)), eps=1e-4, eps2=1e-6)
```

Arguments

xList	A list object containing m data matrices representing m multi-omics data measured in a set of n samples. For each matrix, the rows represent samples, and the columns represent genomic features.
K	An positive integer representing the number of the latent factors used for modeling.
method	Method used for the regularization of model parameters.
min.lambda	The minimum value of the lasso regularization parameter.
max.lambda	The maximum value of the lasso regularization parameter.
lambda.iter	the number of iteration to find an optimal lambda.
EM.iter	The number of iteration for the EM algorithm.
min.shrinkage.rate	The minimum lasso shrinkage rates for multi-omics features.
eps	EM algorithm convergence criterion 1.
eps2	EM algorithm convergence criterion 2.

Value

A list with the following elements.

meanZ	A n x k matrix; the rows represent samples and the columns represent the K factors.
beta	A list object containing m coefficient matrices corresponding to the m data matrices.
lambda	Final lasso regularization parameters used for iCluster modeling.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Ronglai Shen, Sijian Wang, Qianxing Mo. (2013). Sparse Integrative Clustering of Multiple Omics Data Sets. *Annals of Applied Statistics.* 7(1):269-294

See Also

iCluster2b, iClusterPlus2, iClusterBayes

`tune.iClusterBayes` *Integrative clustering of multiple genomic data*

Description

In order to determining the appropriate number of clusters, tune.iClusterBayes calls iClusterBayes function and performs parallel computation for K=1,2,....

Usage

```
tune.iClusterBayes(cpus=6,dt1,dt2=NULL,dt3=NULL,dt4=NULL,dt5=NULL,dt6=NULL,
 type=c("gaussian","binomial","poisson"),
    K=1:6,n.burnin=1000,n.draw=1200,prior.gamma=rep(0.1,6),
 sdev=0.5,beta.var.scale=1,thin=1,pp.cutoff=0.5)
```

Arguments

`cpus`	Number of CPU used for parallel computation. If possible, let it be equal to the number of Ks.
`dt1`	Data set 1 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt2`	Data set 2 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt3`	Data set 3 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt4`	Data set 4 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt5`	Data set 5 - a matrix with rows and columns representing samples and genomic features, respectively.
`dt6`	Data set 6 - a matrix with rows and columns representing samples and genomic features, respectively.
`type`	Data type corresponding to dt1-6, which can be gaussian, binomial, poisson.
`K`	A vector. Each element is the number of eigen features. Given k, the number of cluster is k+1.
`n.burnin`	Number of MCMC burnin.
`n.draw`	Number of MCMC draw.
`prior.gamma`	Prior probability for the indicator variable gamma of each data set.
`sdev`	Standard deviation of random walk proposal for the latent variable.
`beta.var.scale`	A positive value to control the scale of covariance matrix of the proposed beta.
`thin`	A parameter to thin the MCMC chain in order to reduce autocorrelation. Discard all but every 'thin'th sampling values. When thin=1, all sampling values are kept.
`pp.cutoff`	Posterior probability cutoff for the indicator variable gamma. The BIC and deviance ratio will be calculated by setting parameter beta to zero when the posterior probability of gamma <= cutoff.

Value

A list named 'fit'. fit[[i]] is an object return by iClusterBayes, corresponding to the ith element in K. Each component of fit has the following elements.

`alpha`	Intercept parameter.
`beta`	Information parameter.
`beta.pp`	Posterior probability of beta. The higher the beta.pp, the more likely the beta should be included in the model.
`gamma.ar`	Acceptance ratio for parameter gamma.
`beta.ar`	Acceptance ratio for parameter beta.
`Z.ar`	Acceptance ratio for the latent variable.
`clusters`	Cluster assignment.
`centers`	Cluster center.
`meanZ`	Latent variable.
`BIC`	Bayesian information criterion.
`dev.ratio`	See dev.ratio defined in glmnet package.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics 19(1):71-86.

See Also

`iClusterBayes`,`plotHMBayes`,`iClusterPlus`,`tune.iClusterPlus`,`plotHeatmap`

Examples

```
### see the users' guide iManul.pdf
```

`tune.iClusterPlus` *Integrative clustering of multiple genomic data*

Description

Given multiple genomic data (e.g., copy number, gene expression, DNA methylation) measured in the same set of samples, tune.iClusterPlus uses a series of lambda values to fit a regularized latent variable model based clustering that generates an integrated cluster assignment based on joint inference across data.

Usage

```
tune.iClusterPlus(cpus=8,dt1,dt2=NULL,dt3=NULL,dt4=NULL,
  type=c("gaussian","binomial","poisson","multinomial"),
  K=2,alpha=c(1,1,1,1),n.lambda=NULL,scale.lambda=c(1,1,1,1),
  n.burnin=200,n.draw=200,maxiter=20,sdev=0.05,eps=1.0e-4)
```

Arguments

`cpus`	Number of CPU used for parallel computation.
`dt1`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt2`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt3`	A data matrix. The rows represent samples, and the columns represent genomic features.
`dt4`	A data matrix. The rows represent samples, and the columns represent genomic features.
`type`	data type, which can be "gaussian","binomial","poisson", and"multinomial".
`K`	The number of eigen features. Given K, the number of cluster is K+1.
`alpha`	Vector of elasticnet penalty terms. At this version of iClusterPlus, elasticnet is not used. Therefore, all the elements of alpha are set to 1.
`n.lambda`	Number of lambda are tuned.
`scale.lambda`	A value between (0,1); the actual lambda values will be scale.lambda multiplying the lambda values of the uniform design.
`n.burnin`	Number of MCMC burnin.
`n.draw`	Number of MCMC draw.
`maxiter`	Maximum iteration for the EM algorithm.
`sdev`	standard deviation of random walk proposal.
`eps`	EM algorithm convergence criterion.

Value

A list with the two elements 'fit' and 'lambda', where fit itself is a list and lambda is a matrix. Each row of lambda is the lambda values used to fit iClusterPlus model. Each component of fit is an object return by iClusterPlus, one-to-one corresponding to the row of lambda. Each component of fit has the following objects.

`alpha`	Intercept parameter for the genomic features.
`beta`	Information parameter for the genomic features. The rows and the columns represent the genomic features and the coefficients for the latent variable, respectively.
`clusters`	Cluster assignment.
`centers`	Cluster centers.
`meanZ`	Latent variable.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>, Ronglai Shen <shenr@mskcc.org>

References

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. (2012). Pattern discovery and cancer gene identification in integrated cancer genomic data. Proc. Natl. Acad. Sci. USA 110(11):4245-50.

See Also

`plotiCluster`,`iClusterPlus`,`iCluster2`,`iCluster`, `compute.pod`

Examples

```
### see the users' guide iManul.pdf
```

`UM` *Uveal Melanoma data*

Description

This is a subset of the uveal melanoma (UM) multi-omics data from the cancer genome atlas (TCGA) study (2018), which were re-analyzed by Mo et al. (2021).

Usage

```
data(UM)
```

Format

Data matrices of somatic mutation, DNA copy number, methylation and mRNA expression for 80 UM primary samples.

Details

The TCGA UM multi-omics data (version 2016_01_28) were obtained from the Firebrowse portal (http://firebrowse.org/, accessed on 19 December 2018). The level 3 multi-omics data were processed for iCluster analysis, which were detailed in the Materials and Methods of Mo et al. (2021).

Value

`mut02`	Somatic mutation data matrix with 0 representing wild type and 1 representing somatic mutation. Genes with mutation rate >= 2% in the 80 samples are kept in the data matrix.
`cn`	Copy number regions, which were generated by merging the log2 ratios of chromosome segments using the CNregions function.
`methy25`	Methylation data matrix made of the top 25% most variable genes.
`mrna25`	mRNA expression data matrix made of the top 25% most variable genes.
`methy25Anno`	Annotation for the genes in the methylation data matrix methy25.
`clin4`	Clinical data of the 80 UM samples.

References

Robertson, A.G.; Shih, J.; Yau, C.; Gibb, E.A.; Oba, J.; Mungall, K.L.; Hess, J.M.; Uzunangelov, V.;Walter, V.; Danilova, L.; et al. Integrative Analysis Identifies Four Molecular and Clinical Subsets in Uveal Melanoma. 2018. *Cancer Cell* 33 (1), 151.

Mo, Q; Wan, L; Schell, MJ; Jim, H; Tworoger, SS; G Peng, G. Integrative analysis identifies multi-omics signatures that drive molecular classification of uveal melanoma. 2021. *Cancers* 13 (24), 6168

`utility` *Utility functions for iClusterPlus package*

Description

Some utility functions for processing the results produced by iClusterPlus methods.

Usage

```
getBIC(resultList)
getDevR(resultList)
getClusters(resultList)
iManual(view=TRUE)
```

Arguments

resultList	A list object as shown in the following example.
view	A logical value TRUE or FALSE

Value

getBIC	produce a matrix containing the BIC value for each lambda and K; the rows correspond to the lambda (vector) and the columns correspond to the K latent variables.
getDevR	produce a matrix containing the deviance ratio for each lambda and K; the rows correspond to the lambda (vector) and the columns correspond to the K latent variables.
getClusters	produce a matrix containing the cluster assigments for the samples under each K; the rows correspond to the samples; the columns correspond to the K latent variables.
iManual	Open the iClusterPlus User's Guide.

Author(s)

Qianxing Mo <qianxing.mo@moffitt.org>

References

Qianxing Mo, Sijian Wang, Venkatraman E. Seshan, Adam B. Olshen, Nikolaus Schultz, Chris Sander, R. Scott Powers, Marc Ladanyi, and Ronglai Shen. (2012). Pattern discovery and cancer gene identification in integrated cancer genomic data. Proc. Natl. Acad. Sci. USA (invited revision).

See Also

tune.iClusterPlus, iClusterPlus, iCluster2

Examples

```
### see the users' guide iManual.pdf

#data(simuResult)
#BIC = getBIC(simuResult)
#devR = getDevR(simuResult)
#clusters = getClusters(simuResult)
```

`variation.hg18.v10.nov.2010` *Human genome variants of the NCBI 36 (hg18) assembly*

Description

Human genome variants of the NCBI 36 (hg18) assembly

Usage

```
data(variation.hg18.v10.nov.2010)
```

Format

data frame

Value

`variation.hg18.v10.nov.2010`
: Human genome variants of the NCBI 36 (hg18) assembly

References

http://projects.tcag.ca/variation/tableview.asp?table=DGV_Content_Summary.txt

Index

∗ **datasets**
breast.chr17, 2
coord, 5
gbm, 5
glp, 6
UM, 31
variation.hg18.v10.nov.2010, 34
∗ **models**
CNregions, 3
compute.pod, 4
iCluster, 6
iCluster2, 8
iCluster2b, 10
iClusterBayes, 13
iClusterBayes2, 14
iClusterPlus, 16
iClusterPlus2, 18
pcaVarPlot, 19
plotHeatmap, 20
plotHMBayes, 22
plotiCluster, 23
plotRI, 24
tune.iCluster2, 25
tune.iCluster2b, 26
tune.iClusterBayes, 28
tune.iClusterPlus, 30
utility, 32

breast.chr17, 2, *3*, *7*

clin4 *(*UM*)*, 31
cn *(*UM*)*, 31
CNregions, 3
compute.pod, *3*, 4, *7*, *9*, *17*, *24*, *31*
coord, 5

gbm, 5
getBIC *(*utility*)*, 32
getClusters *(*utility*)*, 32
getDevR *(*utility*)*, 32
glp, 6

iCluster, *3, 4*, 6, *17*, *21*, *24, 25*, *31*
iCluster2, *4*, 8, *26*, *31*, *33*
iCluster2b, 10, *21*, *27*
iClusterBayes, *11*, 13, *19, 20*, *23*, *27*, *29*
iClusterBayes2, 14
iClusterPlus, *3*, *9*, *14*, 16, *19*, *29*, *31*, *33*
iClusterPlus2, *11*, *16*, 18, *20, 21*, *27*
iManual *(*utility*)*, 32

methy25 *(*UM*)*, 31
methy25Anno *(*UM*)*, 31
mrna25 *(*UM*)*, 31
mut02 *(*UM*)*, 31

pcaVarPlot, 19
plotHeatmap, *14*, *16*, *20*, 20, *23*, *29*
plotHMBayes, *14*, *16*, 22, *29*
plotiCluster, *3, 4*, *7*, *9*, *17*, 23, *31*
plotRI, 24

tune.iCluster2, 25
tune.iCluster2b, *11*, *19, 20*, 26
tune.iClusterBayes, *14*, 28
tune.iClusterPlus, *14*, *29*, 30, *33*

UM, 31
utility, 32

variation.hg18.v10.nov.2010, 34