Package ‘CSclone’

November 12, 2016

Type Package

Title Bayesian Nonparametric Modeling in R

Version 1.0

Date 2016-11-10

Author Peter Wu <peter123wu0@gmail.com>

Maintainer Peter Wu <peter123wu0@gmail.com>

Description Germline and somatic locus data which contain the total read depth and B allele read depth using Bayesian model (Dirichlet Process) to cluster. Meanwhile, the cluster model can deal with the SNVs mutation and the CNAs mutation.

License GPL (>= 2)

Depends lpSolve, mcclust, moments, DNAcopy, stats, R (>= 2.10)

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-11-12 00:52:58

R topics documented:

adjust.mpear *Adjust MPEAR*

Description

adjust.mpear is modified the mcclust package to suit our model.

Usage

```
adjust.mpear(psm, method = "ward.D2", max.k = NULL)
```

Arguments

`psm`	is posterior similarity matrix with entries between 0 and 1 and 1's on the diagonals which is distance matrix.
`method`	is clustering method. We offer many kinds of method, like as kmeans and hierarchical clustering. The default is ward.D2 of hierarchical clustering.
`max.k`	is number with the user has the professional judgment to consider the max group and the default is NULL.

Details

adjust.mpear is modified the mcclust package to suit our model. And the function use distance matrix by criteria of MPEAR to cluster the data.

Value

cl is vector with the clustering result.

value is number with MPEAR.

method is clustering method.

Author(s)

Peter Wu (peter123wu0@gmail.com)

References

Fritsch, A. and Ickstadt, K. Improved criteria for clustering based on the posterior similarity matrix. Bayesian analysis 2009;4(2):367-391.

CSclone *Clustering model which contains the CNAs mutation and SNVs mutation.*

Description

CSclone is the main function of the package and a clustering model which contains the CNAs mutation and SNVs mutation.

Usage

```
CSclone(somatic.id, DNAcopy.object, mcmc = list(nburn = 5000, nsave = 10000,
  nskip = 1, ndisplay = 1000), y, set = 100, alpha = 1, max.k = NULL,
  method = "ward.D2", prior = c(1, 1))
```

Arguments

`somatic.id`	is a vector of s observiations and mark which loci has somatic mutation.
`DNAcopy.object`	is a list and output of the DNAcopy package.
`mcmc`	is a list giving the MCMC parameters. The list must include the following integers: nburn giving the number of burn-in scans, nskip giving the thinning interval, nsave giving the total number of scans to be saved, and ndisplay giving the number of saved scans to be displayed on screen (the function reports on the screen when every ndisplay iterations have been carried out) and default is list(nburn=5000,nsave=10000,nskip=1,ndisplay=1000).
`y`	is a matrix giving the binomial data. The first column is B allele read depth and the second column is total read depth.
`set`	is number that you can reproduce the simulation result and default is 100.
`alpha`	is a number giving the concentration parameter of Dirichlet process and default is 1.
`max.k`	is a number giving limit the maximum cluster number and default is NULL.
`method`	is the agglomeration method to be used and the method is used to result of Hamming distance. The default is ward.D2.
`prior`	is a vector has tow number which is the parameter of Beta distribution and the default is (1,1).

Details

CSclone is a two steps model. The first step is clustering the non-CNAs somatic mutations by DPMM with Binomial distribution and the posterior distribution is Beta distribution. The second step classifies the CNAs mutation and the somatic mutation with CNAs mutation. There are an example and the dataset is small case in order to run quickly.

Value

cluster is a vector giving the group result of the somatic mutation.

segment is a matrix. The first column is chromosome, the second column is starting position, the third column is end position, the fourth column is copy number, the fifth column is proportion of mutation, the sixth is number of loci, the seventh column is starting loci number, and the eighth column is end loci number.

mcmc is a list giving the MCMC parameters.

alpha is number giving the concentration parameter of Dirichlet process.

y is a matrix giving the binomial data.

group.prop is number giving the predict the proportion of group.

SNV is matrix giving the somatic mutation of every loci. The first column is B allele read depth, the second column is total read depth, the third column proportion of somatic mutation.

Author(s)

Peter Wu (peter123wu0@gmail.com)

Examples

```
mcmc=list(nburn=200,nsave=500,nskip=1,ndisplay=1000)
p=c(0.2,0.4,0.6,0.8)
chrs=rep(c(1:2),times=rep(500,2))
pos=sort(sample(size=1000,x=1:10^7))
pc=sample(rep(p,4*c(0,0,0.5,0.5)))
ps=sample(rep(p,50*c(0.3,0.3,0.2,0.2)))
x=simu.data(n.germline=1000,pc=pc,read=200,ps=ps,dis="Negative binomial",parameter=0.75)
snv.id=x$snv.id
y=x$y
row.names(y)=paste0(chrs,"_",pos)
logR=log(y[,2],base=2)-median(log(y[,2],base=2))
data=data.frame(chr=chrs,pos=pos,logR=logR)
rownames(data)=paste0("SNP",1:nrow(data))
CNA.object=CNA(data$logR,data$chr,data$pos,data.type="logratio",sampleid="test")
smoothed.CNA.object=smooth.CNA(CNA.object)
DNAcopy.object=segment(smoothed.CNA.object, undo.splits = "sdundo",undo.SD = 3, verbose = 1)
fit=CSclone(somatic.id=snv.id,DNAcopy.object=DNAcopy.object,mcmc=mcmc,y=y)
result=fit$cluster
tf_similar(real=ps,cluster=result)
```

`Hamming` *Hamming distance*

Description

Hamming is computing the co-occurrence matrix.

Usage

```
Hamming(k.matrix)
```

Arguments

`k.matrix`	is matrix with the result-retaining of MCMC.

Details

Hamming use the result-retaining of MCMC to compute the co-occurrence matrix. And we use the co-occurrence matrix to cluster which the criteria is MPEAR.

Value

Hamming.matrix is the co-occurrence matrix and the idea is Hamming distance.

pro.matrix is the standardization of co-occurrence matrix.

data is data.frame with matrix. The purpose is constructing the heat map.

Author(s)

Peter Wu (peter123wu0@gmail.com)

`l.s` *Linear relation for SNVs*

Description

l.s is the linear relation between B allele frequence and the proportion of SNVs mutation(PS).

Usage

```
l.s(cnt, cnb1, cnb2, pc = 0, ps = NULL, baf = NULL)
```

Arguments

`cnt`	is number with the total copy number after the CNAs mutation.
`cnb1`	is number with the B copy number of two types of periods. The first period, the proportion of CNAs mutation is more than the proportion of SNVs mutation then cnb1 is B copy number after SNVs mutation. The second period, the proportion of CNAs mutation is less than the proportion of SNVs mutation then cnb1 is B copy number between the SNVs mutation and CNAs mutation.
`cnb2`	is number with the B copy number of two types of periods. The first period, the proportion of CNAs mutation is less than the proportion of SNVs mutation then cnb1 is B copy number after CNAs mutation. The second period, the proportion of CNAs mutation is more than the proportion of SNVs mutation then cnb1 is B copy number between the CNAs mutation and SNVs mutation.

`pc` is number with the proportion of CNAs mutation and the default is 0.

`ps` is number with the proportion of SNVs mutation and the default is NULL.

`baf` is number with the B allele frequence and the default is NULL.

Details

l.s is bidirection function. The first function is given the proportion of SNVs mutation(ps) to predict the B allele frequence(baf). The second function is given the B allele frequence(baf) to predict the proportion of SNVs mutation(ps).

Value

baf is number with B allele frequence(baf) if the input is given the proportion of SNVs mutation(ps).

ps is number with the proportion of SNVs mutation(ps) if the input is given the B allele frequence(baf).

Author(s)

Peter Wu (peter123wu0@gmail.com)

`simu.data` *Simulate data*

Description

simu.data is generating the data which simulates the locus.

Usage

```
simu.data(n.germline = 10000, segment.length = 100, seed = NULL, pc,
  read = 100, ps, dis = "Normal", parameter = NULL)
```

Arguments

`n.germline` is number with the total number of generating data which contains non-CNAs and non-SNVs. The default is 10000.

`segment.length` is number with the number of loci of every segment and the number is common divisor of n.germline. The default is 100.

`seed` is number that you can reproduce the simulation result and default is NULL.

`pc` is vector with the proportion of CNAs mutation of every segment.

`read` is number with the standard total read depth.

`ps` is vector with the proportion of SNVs mutation.

`dis` is distribution with generating the data and the default is Normal distribution. We provide Poisson and Negative binomial distribution to choose.

`parameter` is number and default is NULL. If dis is Normal and parameter is not NULL, then the standard deviation is parameter x mean. If dis is Normal and parameter is NULL, then the standard deviation is mean. If dis is Negative binomial and parameter is NULL, then the parameter is second parameter. If dis is Negative binomial and parameter NULL, then the second parameter is 0.5.

Details

The simu.data can consider not only SNVs mutation but also CNAs mutation and generate the data which contains germline data and mutation data.

Value

y is n x 2 matrix. The first column is the B allele read depth and the second column is the total read depth.

snv.id is vector which denotes which loci with SNVs mutation.

cnt.id is A x B matrix which denotes which loci with CNAs mutation.. The A is segment.length and the B is the number of segment with CNAs mutation.

cnv is n x 4 matrix. The first column is the total copy number after CNAs mutation, the second column is the cnb1, the third column is the cnb2, and the fourth column is the proportion of CNAs mutation. The cnb1 and the cnb2 has detailed explanation at l.s function.

Author(s)

Peter Wu (peter123wu0@gmail.com)

`tf_similar` *True or false of similar*

Description

tf_similar is comparing the similarity of different group.

Usage

```
tf_similar(real, cluster)
```

Arguments

`real` is vector with the real group number. If we don't have the real group number, we also can input the result of some clustering method.

`cluster` is vector with the result of clustering.

Details

tf_similar is comparing the similarity of different group. So the main purpose is checking the quality of clustering method. We show the rand index(RI), weight rand index(WRI), and adjust rand index(ARI) to compare. In order to compute the three kinds of index, we count the TT, TF, FT, FF, and the relation weight. The details of the function of the function are in paper of Tumor subclones detection with Dirichlet Process Mixture Model.

Value

result is data.frame with 10 columns which contains the TT, TF, FT, FF, rand index(RI), weight rand index(WRI), adjust rand index(ARI), the weighted TT, the weighted TF, and the weight FT.

weight is data.frame with 3 columns which contains the weight about the TT, TF, and FT.

Author(s)

Peter Wu (peter123wu0@gmail.com)

Index