Package ‘MultiCNVDetect’

February 19, 2015

Type Package
Title Multiple Copy Number Variation Detection
Version 0.1-1
Date 2012-10-07
Author Hao Lin, Qiang Kou
Maintainer Hao Lin <sky_lin@163.com>
Description This package provides a tool for analysis of multiple CNV.
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-05-01 12:12:45

R topics documented:

armijo_rule ......................................................... 2
bladder ............................................................. 2
calculate_direction ............................................... 3
delta_k .............................................................. 3
Fx ................................................................. 4
las_bycgd ......................................................... 5
MultiCNVDetect .................................................. 6
norm_2 ............................................................ 7
P ................................................................. 7

Index 8
Calculate alpha using armijo_rule in BCGD algorithm

Description

In BCGD algorithm, armijo_rule is used to get the alpha in each iterative step.

Usage

armijo_rule(Y, X, D, lambda1, lambda2, alpha)

Arguments

Y A n*m matrix, in the multiple copy number variation detection problem, n denotes #(n) samples, while m denotes #(m) probs.
X A n*m matrix, original estimator of Y, give X as the same size of Y, often let X=Y.
D A n*m matrix, original direction in BCGD algorithm.
lambda1 A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
lambda2 A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.
alpha Original alpha used in the first step.

Value

Returns an object of scale.

Bladder Data

Description

The bladder data contains CGH array data of 57 bladder tumor samples, and each sample has 2143 probs.

Usage

data(bladder)

Source

http://microarrays.curie.fr/publications/oncologie_moleculaire/bladder_TCM/

References

**calculate_direction**  
*Calculate direction in each iterative step in BCGD algorithm*

**Description**
In BCGD algorithm, we should calculate the direction $d$ in each iterative step. According to the multiple copy number variation detection problem, we solve it using chasing method.

**Usage**

```
calculate_direction(Y, X, lambda1, lambda2)
```

**Arguments**
- **Y**: A $n \times m$ matrix.
- **X**: A $n \times m$ matrix, should be the same size of Y.
- **lambda1**: A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
- **lambda2**: A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.

**Value**
It returns a $n \times m$ matrix.

---

**delta_k**  
*Calculate delta used in armijo rule.*

**Description**
We choose alpha by armijo rule. It satisfies $F(x+\alpha*d) \leq \alpha*\sigma*\delta + F(x)$, where $\alpha$ is the largest element of $\alpha_{init}*\beta^j, (j=0, 1, ...)$, In BCGD algorithm, the author defines delta by a function. (See more details in Reference)

**Usage**

```
delta_k(Y, X, D, lambda1, lambda2)
```
Arguments

Y A n*m matrix.
X A n*m matrix, should be the same size of Y.
D A n*m matrix, the direction in each step, can be calculated by function calculate_direction.
lambda1 A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
lambda2 A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.

References


Fx

Calculate the target function F(x) in BCGD algorithm

Description

F(x)=f(x)+cP(x), the function just calculates F(x) when given x=x0. (f is smooth, P is a proper, convex, lower semicontinuous function, c is the tuning parameter, c>0, see more details in references)

Usage

Fx(Y, X, lambda1, lambda2)

Arguments

Y A n*m matrix.
X A n*m matrix, the same size of Y.
lambda1 A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
lambda2 A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.

References

Description

Set up a model for multiple copy number variation detection with fused lasso and group lasso, the solve it using Block Coordinate Gradient Descent algorithm.

Usage

`las_bycgd(Y, X = Y, lambda1 = 1, lambda2 = 1)`

Arguments

- **Y**: A n*m matrix, in copy number variation detection problem, n denotes #(n) samples and m denotes #(m) probs.
- **X**: A n*m matrix, the initial estimator of Y, with the same size of Y, often given X=Y.
- **lambda1**: A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
- **lambda2**: A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.

References


Examples

```r
# simulated data: 5 samples, 980 probs.
sigma <- 0.4; a <- 0.5; i <- matrix(1:980, 980, 1); n <- 5
j <- matrix(1:n, ncol = 1)
g <- function(i) 0.25 * sigma * sin(a * pi * i)
err <- function(i){
  set.seed(i);
  return(rnorm(980, 0, sigma))
}
y <- matrix(, n, 980); set.seed(123)
a1 <- matrix(rnorm(n, 0, 1), n, 300);
a2 <- matrix(rnorm(n, 0, 1), n, 380);
y[, 1:300] <- a1
y[, 301:480] <- -0.8
y[, 481:980] <- -0.2
y15 <- y + t(apply(j, 1, err)) + apply(j, 1, g)
rm(j, i, a, n, sigma, err, g, a1, a2, y)
beta15 <- las_bycgd(y15)
plot(y15[, 1], col = 2, xlab = "genome order", ylab = "log2 ratio",
     main = "n=15 (the first sample)", ylim = c(-1.5, 2))
lines(beta15[, 1], col = 4)
```
MultiCNVDetect  

Multiple Copy number variation Detection

Description

Detects copy number variation regions with fused lasso and group lasso regression, solved by BCGD algorithm.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>MultiCNVDetect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.1.0</td>
</tr>
<tr>
<td>Date</td>
<td>2012-10-07</td>
</tr>
<tr>
<td>License</td>
<td>GPL(&gt;=2.0)</td>
</tr>
</tbody>
</table>

The most important function is las_bycgd that detects CNV regions.

Author(s)

Hao Lin, Qiang Kou.

Maintainer: Hao Lin <sky_linhao@163.com>

Examples

```r
sigma<-0.4; a<-0.5; i<-matrix(1:980,980,1); n<-15
J<-matrix(1:n,ncol=1)
g<-function(i) 0.25*sigma*sin(a*pi*i)
err<-function(i){
  set.seed(i);
  return(rnorm(980,0,sigma))
}
y<-matrix(,n,980); set.seed(123)
a1<-matrix(rnorm(n,0,1),n,300);
a2<-matrix(rnorm(n,0,1),n,380);
y[,1:300]<-0.2
y[,301:480]<-0.8
y[,481:980]<--0.2
y15<-y+t(apply(J,1,err))+apply(J,1,g)
rm(J,i,a,n,sigma,err,g,a1,a2,y)
beta15<-las_bycgd(y15);
plot(y15[1,],col=2,xlab="genome order",ylab="log2 ratio",
     main="n=15(the first sample)", ylim=c(-1.5,2))
lines(beta15[1,],col=4)
```
Compute the 2-Norm of a vector

**Description**

Calculate the norm of a vector \( x \), the norm can only be the "2"-norm.

**Usage**

\[
\text{norm}_2(x)
\]

**Arguments**

- \( x \)  
  - \( x \) is a vector.

**P**

Compute \( P(x) \) in BCGD algorithm

**Description**

In BCGD algorithm, the target function is \( F(x) = f(x) + cP(x) \). Function \( P \) calculates \( P(x) \) given \( x = x_0 \).

**Usage**

\[
P(X, \lambda_1, \lambda_2)
\]

**Arguments**

- \( X \)  
  - \( n \times m \) matrix.
- \( \lambda_1 \)  
  - A real number larger than 0, the first penalty used in the model, controls the overall copy number alteration amount of the target chromosome.
- \( \lambda_2 \)  
  - A real number larger than 0, the second penalty used in the model, controls the frequency of the alterations in the target region.
Index

*Topic **BCGD**
  las_bycgd, 5
*Topic **CNV, BCGD, fused lasso, group lasso**
  MultiCNVDetect, 6
*Topic **armijo_rule**
  armijo_rule, 2
*Topic **calculate_direction**
  calculate_direction, 3
*Topic **datasets**
  bladder, 2
*Topic **group lasso**
  las_bycgd, 5
*Topic **norm**
  norm_2, 7

armijo_rule, 2
bladder, 2
calculate_direction, 3
delta_k, 3
Fx, 4
las_bycgd, 5
MultiCNVDetect, 6
norm_2, 7
P, 7