Package 'cghFLasso'

February 19, 2015

Version 0.2-1

Date 2009-06-01

Title Detecting hot spot on CGH array data with fused lasso regression.

Author Nicholas Austin Johnson <nickaj@stanford.edu>, Robert Tibshirani <tibs@stat.stanford.edu>, Pei Wang

<pwang@fhcrc.org>.

Maintainer Pei Wang <pwang@fhcrc.org>

Description Spatial smoothing and hot spot detection using the fused lasso regression

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-06-07 19:19:51

NeedsCompilation yes

R topics documented:

CGH *Example CGH Array Data*

Description

A list of four components (an example CGH array data for package cghFLasso)

Usage

```
data(CGH)
```

Format

Both NormalArray and DiseaseArray are muneric matrix of 2270 rows and 3 columns: each row corresponds to one gene/clone, and each column corresponds to one CGH array. The value of each entry is the log fluorescence ratio resulted from the CGH experiment. The order of the genes/clones in the rows is the same as the order of the genes/clones on the genome. chromosome and nucposition provide chromosome number and nucleotide position for each gene/clone. \ GBM.y is a numeric vertor of length 990, providing CGH measurement of one pseudo chromosome.

References

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani, "A method for calling gains and losses in array CGH data", Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

cghFLasso *A function to call alteration for CGH arrays using the fused lasso regression*

Description

A function to call alteration for CGH arrays using the fused lasso regresssion

Usage

```
cghFLasso(CGH.Array, chromosome=NULL, nucleotide.position=NULL, FL.norm=NULL, FDR=NULL, filter=NULL,
```

Arguments

CGH.Array	numeric vector or matrix. It's the result of one or mutiiple CGH experiments. Each column is the log2 ratios returned from one array experiment and is ordered according to the gene/clones' position on the genome. Missing value should be coded as NA.
chromosome	numeric vector. Length should be the same as the row number of CGH.Array. It's the chromosome number of each gene/clone. If no value is specified, the arrays will be treated as one chromosome.
nucleotide.position	numeric vector. Length should be the same as the row number of CGH.Array. It's the nucleotide position of each gene/clone. This information is used mainly for plot. If no value is specified, the program will make genes/clones equally spaced on the genome.
FL.norm	numeric vector or matrix. Smoothed result of the reference arrays. Set to NULL (default) if the reference arrays are not available.
FDR	numeric value (between 0 and 1). User can use this option to control False Discovery Rate of the results. If not specified, the function will return the raw output of fused lasso regression on the target array.
filter	numeric vector. Length should be the same as the row number of CGH.Array. Each element takes value 1 or 0. Value 1 means that the corresponding gene/clone will be filter away in ahead of the analysis. If no filter is specified, the program will process all the genes/clones in the input arrays.
missing.PlugIn	Bollen value. If its value is TRUE, the missing values will be replaced with local averages.
smooth.size	numeric value, specifying the window size for carrying out the local average. The average is used only to replace the missing values.

Details

cghFLasso calls copy number alterations for CGH arrays using the fused lasso regression. The dynamic programming algorithm developed by N.A.Johnson is used for the model fitting.

Value

an object of class cghFLasso with components

Esti.CopyN	data matrix reporting the estimated DNA copy numbers for seleted genes/clones of all the samples.
CGH.Array	data matrix reporting the raw CGH array measurements for selected genes/clones of all the samples.
chromosome, nucleotide.position	numeric vectors reporting the chromosome numbers and the nucleotide positions of selected genes/clones. If filter=NULL, these are the same as the input chromosome and nucleotide.position.
FDR	fdr value specified by the user.

Author(s)

N. A. Johnson, R. Tibshirani and P. Wang

References

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2004) 'Sparsity and smoothness via the fused lasso', J. Royal. Statist. Soc. B. (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani (2005) 'A method for calling gains and losses in array CGH data', Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

J. Friedman, T. Hastie. R. Tibshirani (2007) 'Pathwise coordinate optimization and the fused lasso'.

Examples

```
library(cghFLasso)
data(CGH)

#############
### Example 1: Process one chromosome vector without using normal references.

CGH.FL.obj1<-cghFLasso(CGH$GBM.y)
plot(CGH.FL.obj1, index=1, type="Lines")

#############
### Example 2: Process a group of CGH arrays and use normal reference arrays.

Normal.FL<-cghFLasso.ref(CGH$NormalArray,  chromosome=CGH$chromosome)
Disease.FL<-cghFLasso(CGH$DiseaseArray, chromosome=CGH$chromosome, nucleotide.position=CGH$nucposition, FL.norm

###  Plot for the first arrays
i<-1
plot(Disease.FL, index=i, type="Single")
title(main=paste("Plot for the ", i ,"th BAC array", sep=""))

### Consensus plot
plot(Disease.FL, index=1:4, type="Consensus")
title(main="Consensus Plot for 4 BAC arrays")

### Plot all arrays
plot(Disease.FL, index=1:4, type="All")
title(main="Plot for all 4 arrays")

### Report and output
report<-summary(Disease.FL, index=1:4)
print(report)
output.cghFLasso(report, file="CGH.FL.output.txt")
```

`cghFLasso.ref` *A function to process reference CGH arrays using the fused lasso regression*

Description

A function to process reference CGH arrays using the fused lasso regression

Usage

```
cghFLasso.ref(CGH.Array, chromosome=NULL, filter=NULL)
```

Arguments

`CGH.Array`	numeric vector or matrix. It's the result of one or mutiiple CGH experiments. Each column is the log2 ratios returned from one array experiment and is ordered according to the gene/clones' position on the genome. Missing value should be coded as NA.
`chromosome`	numeric vector. Length should be the same as the row number of `CGH.Array`. It's the chromosome number of each gene/clone. If no value is specified, the arrays will be treated as one chromosome.
`filter`	numeric vector. Length should be the same as the row number of `CGH.Array`. Each element takes value 1 or 0. Value 1 means that the corresponding gene/clone will be filter away in ahead of the analysis. If no `filter` is specified, the program will process all the genes/clones in the input arrays.

Details

`cghFLasso.ref` fits fused lasso regression on reference CGH arrays . The dynamic programming algorithm developed by N.A.Johnson is used for the model fitting.

Value

numeric matrix representing the smoothed result of the reference arrays.

Author(s)

N. A. Johnson, R. Tibshirani and P. Wang

References

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2004) 'Sparsity and smoothness via the fused lasso', J. Royal. Statist. Soc. B. (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani (2005) 'A method for calling gains and losses in array CGH data', Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

J. Friedman, T. Hastie. R. Tibshirani (2007) 'Pathwise coordinate optimization and the fused lasso'.

Examples

```
library(cghFLasso)
data(CGH)

#############
### Example 1: Process one chromosome vector without using normal references.

CGH.FL.obj1<-cghFLasso(CGH$GBM.y)
plot(CGH.FL.obj1, index=1, type="Lines")

#############
### Example 2: Process a group of CGH arrays and use normal reference arrays.

Normal.FL<-cghFLasso.ref(CGH$NormalArray,  chromosome=CGH$chromosome)
Disease.FL<-cghFLasso(CGH$DiseaseArray, chromosome=CGH$chromosome, nucleotide.position=CGH$nucposition, FL.norm

###  Plot for the first arrays
i<-1
plot(Disease.FL, index=i, type="Single")
title(main=paste("Plot for the ", i ,"th BAC array", sep=""))

### Consensus plot
plot(Disease.FL, index=1:4, type="Consensus")
title(main="Consensus Plot for 4 BAC arrays")

### Plot all arrays
plot(Disease.FL, index=1:4, type="All")
title(main="Plot for all 4 arrays")

### Report and output
report<-summary(Disease.FL, index=1:4)
print(report)
output.cghFLasso(report, file="CGH.FL.output.txt")
```

`output.cghFLasso` *Output gain/loss calls by cghFLasso*

Description

Output gain/loss calls by cghFLasso.

Usage

```
output.cghFLasso(summary.obj, file, gene.info=NULL)
```

Arguments

`summary.obj`	an object of class `summary.cghFLasso`, usually, a result of a call to `summary.cghFLasso`.
`file`	a character specifying the file name to output the data.
`gene.info`	matrix. Additional gene/clone annotation to output.

Details

Output gain/loss calls by cghFLasso.

Value

No return value.

Author(s)

N. A. Johnson, R. Tibshirani and P. Wang

References

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2004) 'Sparsity and smoothness via the fused lasso', J. Royal. Statist. Soc. B. (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani (2005) 'A method for calling gains and losses in array CGH data', Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

J. Friedman, T. Hastie. R. Tibshirani (2007) 'Pathwise coordinate optimization and the fused lasso'.

Examples

```
library(cghFLasso)
data(CGH)

#############
### Example 1: Process one chromosome vector without using normal references.

CGH.FL.obj1<-cghFLasso(CGH$GBM.y)
plot(CGH.FL.obj1, index=1, type="Lines")

#############
### Example 2: Process a group of CGH arrays and use normal reference arrays.

Normal.FL<-cghFLasso.ref(CGH$NormalArray,  chromosome=CGH$chromosome)
Disease.FL<-cghFLasso(CGH$DiseaseArray, chromosome=CGH$chromosome, nucleotide.position=CGH$nucposition, FL.norm

###  Plot for the first arrays
i<-1
plot(Disease.FL, index=i, type="Single")
title(main=paste("Plot for the ", i ,"th BAC array", sep=""))

### Consensus plot
plot(Disease.FL, index=1:4, type="Consensus")
title(main="Consensus Plot for 4 BAC arrays")

### Plot all arrays
plot(Disease.FL, index=1:4, type="All")
title(main="Plot for all 4 arrays")

### Report and output
report<-summary(Disease.FL, index=1:4)
print(report)
output.cghFLasso(report, file="CGH.FL.output.txt")
```

`plot.cghFLasso` *A function to plot gain/loss calls on CGH arrays*

Description

A function to plot gain/loss calls on CGH arrays

Usage

```
plot.cghFLasso(x, index, type="All", centro=NULL, ...)
```

Arguments

x	an object of class cghFLasso (returned by function cghFLasso).
index	numeric vector specifying which arrays to plot.
type	a character specifying the plot type. It should be one of "Lines", "Single", "Consensus", or "All". See details for more information.
centro	numeric vector specifying the centromere positions. If missing, the default centromere value of human genome will be used.
...	further arguments passed to or from other methods.

Details

plot.cghFLasso provides several summary plots for cghFLasso result object. If type="Lines", it plots the raw CGH measurements and the gain/loss results for the one selected array along the genome (no seperation of chromosomes). If type="Single", it plots the raw CGH measurements and the gain/loss results for the one selected array along the genome (different chromosomes are ploted in different lines). If type="Consensus", it plots the percentages of the gain/loss calls across a group of samples along the genome. If type="All", it plots the results of all selected arrays in one figure.

Value

It returns the type of the plot.

Author(s)

N. A. Johnson, R. Tibshirani and P. Wang

References

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2004) 'Sparsity and smoothness via the fused lasso', J. Royal. Statist. Soc. B. (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani (2005) 'A method for calling gains and losses in array CGH data', Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

J. Friedman, T. Hastie. R. Tibshirani (2007) 'Pathwise coordinate optimization and the fused lasso'.

Examples

```
library(cghFLasso)
data(CGH)

#############
### Example 1: Process one chromosome vector without using normal references.
```

```
CGH.FL.obj1<-cghFLasso(CGH$GBM.y)
plot(CGH.FL.obj1, index=1, type="Lines")

#############
### Example 2: Process a group of CGH arrays and use normal reference arrays.

Normal.FL<-cghFLasso.ref(CGH$NormalArray,  chromosome=CGH$chromosome)
Disease.FL<-cghFLasso(CGH$DiseaseArray, chromosome=CGH$chromosome, nucleotide.position=CGH$nucposition, FL.norm

###  Plot for the first arrays
i<-1
plot(Disease.FL, index=i, type="Single")
title(main=paste("Plot for the ", i ,"th BAC array", sep=""))

### Consensus plot
plot(Disease.FL, index=1:4, type="Consensus")
title(main="Consensus Plot for 4 BAC arrays")

### Plot all arrays
plot(Disease.FL, index=1:4, type="All")
title(main="Plot for all 4 arrays")

### Report and output
report<-summary(Disease.FL, index=1:4)
print(report)
output.cghFLasso(report, file="CGH.FL.output.txt")
```

`summary.cghFLasso` *Summarizing gain/loss calls by cghFLasso*

Description

'summary' method for class `cghFLasso`.

Usage

```
summary.cghFLasso(object, index,...)
print.summary.cghFLasso(x,...)
```

Arguments

`object`	an object of class `cghFLasso` (returned by function `cghFLasso`).
`index`	numeric vector specifying which arrays to plot.
`x`	an object of class `summary.cghFLasso`, usually, a result of a call to `summary.cghFLasso`.
`...`	further arguments passed to or from other methods.

Details

summary.cghFLasso summarize the gain/loss calls for a group of CGH arrays. It reports the consensus counts of alterations as well as the corresponding FDRs for each gene/clones. It also returns the sample percentage of amplificaiton and deletion at each gene/cloens.

print.summary.cghFLasso outputs a matrix summarizing the gain/loss calls across all the samples for each chromosome. It returns the maximum sample percentage of amplificaiton and deletion on each chromosome respectively. It also reports the genome percentage of alteration for each chromosome (proportion of genes having consensus FDR smaller than 0.05 on the chromosome).

Value

ans<-list(ConsensusCount=ConsensusCount, CC.FDR=CC.FDR, Amp.ConCount=Amp.CC, Del.ConCount=Del.CC, chrom.summary=chrom.summary, sample.summary=sample.summary, Esti.copy=Esti.copy, chromosome=chromosome, nucposi=nucposi)

ConsensusCount	numeric vector with the same length as each CGH array. It reports the number of samples showing copy number alteration at each gene/clones
CC.FDR	numeric vector of the same length of ConsensusCount. It is the estiamted probability of observing the same or higher consensus count by random chance.
Amp.ConCount	numeric vector of the same length of ConsensusCount. It reports the number of samples showing amplificaiton at each gene/clones.
Del.ConCount	numeric vector of the same length of ConsensusCount. It reports the number of samples showing deletion at each gene/clones.
chrom.summary	matrix with four columns. The first column is the chromosome number. The second and third columns are the maximum sample percentages of amplificaitons and deletions on each chromosome respectively. The forth column represents the genome percentage of alterations on each chromosome.
sample.summary	numeric vector with the same lenght of sample numbers. It reports the overall alteration percentage of each sample.
Esti.copy	numeric matrix showing estiamted copy numbers of all genes/clones of all samples.
chromosome	numeric vector with the same length as each CGH array. It's the chromosome number of each gene/clone.
nucposi	numeric vector with the same length as each CGH array. It's the nucletide position of each gene/clone.

Author(s)

N. A. Johnson, R. Tibshirani and P. Wang

References

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2004) 'Sparsity and smoothness via the fused lasso', J. Royal. Statist. Soc. B. (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan and R. Tibshirani (2005) 'A method for calling gains and losses in array CGH data', Biostatistics 2005, 6: 45-58, available at http://www-stat.stanford.edu/~wp57/CGH-Miner/

R. Tibshirani and P. Wang (2007) 'Spatial smoothing and hot spot detection using the Fused Lasso', Biostatistics (In press), available at http://www-stat.stanford.edu/~tibs/research.html.

J. Friedman, T. Hastie. R. Tibshirani (2007) 'Pathwise coordinate optimization and the fused lasso'.

Examples

```
library(cghFLasso)
data(CGH)

#############
### Example 1: Process one chromosome vector without using normal references.

CGH.FL.obj1<-cghFLasso(CGH$GBM.y)
plot(CGH.FL.obj1, index=1, type="Lines")

#############
### Example 2: Process a group of CGH arrays and use normal reference arrays.

Normal.FL<-cghFLasso.ref(CGH$NormalArray,  chromosome=CGH$chromosome)
Disease.FL<-cghFLasso(CGH$DiseaseArray, chromosome=CGH$chromosome, nucleotide.position=CGH$nucposition, FL.norm

###  Plot for the first arrays
i<-1
plot(Disease.FL, index=i, type="Single")
title(main=paste("Plot for the ", i ,"th BAC array", sep=""))

### Consensus plot
plot(Disease.FL, index=1:4, type="Consensus")
title(main="Consensus Plot for 4 BAC arrays")

### Plot all arrays
plot(Disease.FL, index=1:4, type="All")
title(main="Plot for all 4 arrays")

### Report and output
report<-summary(Disease.FL, index=1:4)
print(report)
output.cghFLasso(report, file="CGH.FL.output.txt")
```

Index