Package 'LDcnv'

July 14, 2020

Type Package

Title LDcnv: Integrating Genomic Correlation Structure in Copy Number Variations Detection

Version 1.0

Date 2020-07-14

Author Xizhi Luo, Fei Qin, Guoshuai Cai, Feifei Xiao

Description The detection of copy number variants (CNVs) is identifying mean shift in genetic intensities to locate chromosomal breakpoints. Many segmentation algorithms have been developed with a strong assumption of independent observations in the genetic loci, and they assume each locus has an equal chance to be a breakpoint (i.e., boundary of CNVs). However, this assumption is violated in the genetics perspective due to the existence of correlation among genomic positions such as linkage disequilibrium (LD). To generate more accurate CNVs, we therefore proposed a novel algorithm, LDcnv, that models the CNV data with its biological characteristics relating to genetic correlation (i.e., LD).

R topics documented:

`gausianMixture` *Clustering of CNVs using Expectation-Maximization algorithm*

Description

This function clusters the identified change-points to make final CNV calling.The potential CNV segments between two neighbor candidate change-points are assigned to different copy number states according to the average intensities in the segment intervals. We use three clusters including duplication, normal state and deletion. Each cluster is presented by Gaussian distribution with unknown mean and variance. Expectation-Maximization (EM) algorithm is applied for the mixture of Gaussians to assign each segment to the most probable cluster/state. Two physically linked candidate CNV segments in the same group are merged to one unique CNV segment.

Usage

```
gausianMixture(x, cp, priors, L, st)
```

Arguments

x	the vector of the intensities of markers
cp	the vector of the marker index of the identified change-points
priors	given initial parameters for the EM algorithm
L	repeat times in the EM algorithm. Defaults to 100
st	number of assumed states in the EM algorithm

Value

The return is the clustered CNV segments by presenting the start position and end position using SNP or CNV marker index, and the copy number states. It also returns a vector of final candidates of change-points.

p.final probability of falling into each state for each CNV segment after convergence

mu.final segment means of each state after convergence

cp.final list of change-points after EM algorithm

index.final the bandwidth of change-points

state.new assigned copy number state for each CNV

---

getOneBIC *The modified Bayesian Information Criterion step to remove false positives in change-points*

Description

The modified Bayesian Information Criterion step to remove false positives in change-points

Usage

```
getOneBIC(x, cp, mod = FALSE)
```

Arguments

x	the vector of the intensities of markers
cp	the vector of the marker index of the identified change-points

Value

a list of change-points after filtering false positives

`LDCNVout` *LDCNVout*

Description

This function annotates the identified CNV using the reference map file and output the annotation of all identified CNVs. Each line of the output describes one CNV in nine columns: individual ID; chromosome ID; CNV start marker identifier; CNV start location (in base pair units); CNV end marker identifier; CNV end location (in base pair units); length of CNV (in base pair units); length of CNV(number of markers); copy number states (duplication or deletion).

Usage

```
LDCNVout(
  lrr,
  map,
  h1 = 5,
  h2 = 10,
  h3 = 15,
  L = 100,
  sigma = NULL,
  precise = 10000,
  FINV = FINV,
  alpha = 0.01,
  thre = 10,
  dis.thre = 5,
  outname = outname
)
```

Arguments

`lrr`	the matrix of the lrr intensities. Each column describes a single sample or sequence and each row describes a single marker
`map`	Each line of the map file describes a single marker and must contain exactly 3 columns: chromosome ID; rs# or marker identifier; position (in bp units)
`h1`	the bandwidth 1 for the screening procedure, defaults to 5
`h2`	the bandwidth 2 for the screening procedure, defaults to 10
`h3`	the bandwidth 3 for the screening procedure, defaults to 15
`L`	number of iterations in the EM algorithm for CNV clustering
`alpha`	the significance levels for the test to accept change-points
`thre`	the threshold for CNV length
`dis.thre`	the threshold for distance between CNVs for merging adjacent closely located CNVs
`outname`	name for the output file

Value

This function generates a text file describing all detected CNVs. In addition, it also returns a list of detected change-points for all samples.

cp a list of position index for the final change-points identified by modSaRa

See Also

modifiedSaRa for processing the modified SaRa method

Examples

```
# Input the example data of SNP genotyping data from Affymatrix Human SNP Array 6.0 platform.
# The map file displays annotation of the markers including the chromosome and location
# information of each SNP or CNV marker.
data(example.data.lrr)
data(example.data.baf)
data(example.data.map)
# The following file will be generated: "out.csv"
# This file contains CNV output for each individual.
# Each line represents one CNV detected from one sample or sequence.
# For each line, the individual ID, start position, end position, length and state
# (duplication or deletion) of the CNV will be shown.
out.cp <- cnv.out$cp
# This returns a list of vectors containing detected change-points by modSaRa for each
# sample in the marker name.
```

LDcnv_eCN *LDcnv_eCN This function uses both lrr and baf intensities This function annotates the identified CNV using the reference map file and output the annotation of all identified CNVs. Each line of the output describes one CNV in nine columns: individual ID; chromosome ID; CNV start marker identifier; CNV start location (in base pair units); CNV end marker identifier; CNV end location (in base pair units); length of CNV (in base pair units); length of CNV(number of markers); copy number states (duplication or deletion).*

Description

LDcnv_eCN This function uses both lrr and baf intensities This function annotates the identified CNV using the reference map file and output the annotation of all identified CNVs. Each line of the output describes one CNV in nine columns: individual ID; chromosome ID; CNV start marker identifier; CNV start location (in base pair units); CNV end marker identifier; CNV end location (in base pair units); length of CNV (in base pair units); length of CNV(number of markers); copy number states (duplication or deletion).

Usage

```
LDcnv_eCN(
  lrr,
  baf,
  map,
  alpha = 0.01,
  smooth = TRUE,
  thre = 10,
  dis.thre = 5,
  outname
)
```

Arguments

lrr	the matrix of the lrr intensities. Each column describes a single sample or sequence and each row describes a single marker
baf	the matrix of the baf intensities. Each column describes a single sample or sequence and each row describes a single marker
map	Each line of the map file describes a single marker and must contain exactly 3 columns: chromosome ID; rs# or marker identifier; position (in bp units)
alpha	the significance levels for the test to accept change-points
smooth	specify whether use smooth function to remove outliers of lrr intensities
thre	the threshold for CNV length,default is 10
dis.thre	the threshold for distance between CNVs for merging adjacent closely located CNVs,default is 5
outname	name for the output file

Value

This function generates a text file describing all detected CNVs. In addition, it also returns a list of detected change-points for all samples.

cp a list of position index for the final change-points identified by modSaRa

See Also

modifiedSaRa for processing the modified SaRa method

Examples

```
# Input the example data of SNP genotyping data from Affymatrix Human SNP Array 6.0 platform.
# The map file displays annotation of the markers including the chromosome and location
# information of each SNP or CNV marker.
data(example.data.lrr)
data(example.data.baf)
data(example.data.map)
LDcnv_eCN(lrr = example.data.lrr,baf = example.data.baf,map = example.data.map,outname="out")
# The following file will be generated: "out.csv"
# This file contains CNV output for each individual.
# Each line represents one CNV detected from one sample or sequence.
# For each line, the individual ID, start position, end position, length and state
# (duplication or deletion) of the CNV will be shown.
```

LDcnv_lrr *LDcnv_lrr This function uses lrr intensities This function annotates the identified CNV using the reference map file and output the annotation of all identified CNVs. Each line of the output describes one CNV in nine columns: individual ID; chromosome ID; CNV start marker identifier; CNV start location (in base pair units); CNV end marker identifier; CNV end location (in base pair units); length of CNV (in base pair units); length of CNV(number of markers); copy number states (duplication or deletion).*

Description

LDcnv_lrr This function uses lrr intensities This function annotates the identified CNV using the reference map file and output the annotation of all identified CNVs. Each line of the output describes one CNV in nine columns: individual ID; chromosome ID; CNV start marker identifier; CNV start location (in base pair units); CNV end marker identifier; CNV end location (in base pair units); length of CNV (in base pair units); length of CNV(number of markers); copy number states (duplication or deletion).

Usage

```
LDcnv_lrr(
  lrr,
  map,
  alpha = 0.01,
  smooth = TRUE,
  thre = 10,
  dis.thre = 5,
  outname
)
```

Arguments

lrr	the matrix of the lrr intensities. Each column describes a single sample or sequence and each row describes a single marker
map	Each line of the map file describes a single marker and must contain exactly 3 columns: chromosome ID; rs# or marker identifier; position (in bp units)
alpha	the significance levels for the test to accept change-points
smooth	specify whether use smooth function to remove outliers of lrr intensities
thre	the threshold for CNV length,default is 10
dis.thre	the threshold for distance between CNVs for merging adjacent closely located CNVs,default is 5
outname	name for the output file

Value

This function generates a text file describing all detected CNVs. In addition, it also returns a list of detected change-points for all samples.

cp a list of position index for the final change-points identified by modSaRa

See Also

modifiedSaRa for processing the modified SaRa method

Examples

```
# Input the example data of SNP genotyping data from Affymatrix Human SNP Array 6.0 platform.
# The map file displays annotation of the markers including the chromosome and location
# information of each SNP or CNV marker.
data(example.data.lrr)
data(example.data.map)
LDcnv_lrr(lrr = example.data.lrr,map = example.data.map,alpha=0.01,outname="out1")
```

```
# The following file will be generated: "out.csv"
# This file contains CNV output for each individual.
# Each line represents one CNV detected from one sample or sequence.
# For each line, the individual ID, start position, end position, length and state
# (duplication or deletion) of the CNV will be shown.
```

`modifiedSaRa` *CNV detection processing multiple sequences using the modified SaRa algorithm*

Description

This function runs the modified SaRa algorithm and cluster the change-points to CNVs processing multiple sequences.

Usage

```
modifiedSaRa(
  Y,
  alpha = 0.01,
  h1 = 5,
  h2 = 10,
  h3 = 15,
  L = 100,
  sigma = NULL,
  precise = 10000,
  FINV = FINV
)
```

Arguments

`Y`	the numeric vector of the intensities of markers
`alpha`	the significance levels for the test to accept change-points
`h1`	the bandwidth 1 for the screening procedure, defaults to 5
`h2`	the bandwidth 2 for the screening procedure, defaults to 10
`h3`	the bandwidth 3 for the screening procedure, defaults to 15
`L`	number of iterations in the EM algorithm for CNV clustering
`sigma`	the standard deviation for the intensities between two adjacent change-points, defaults to NULL
`precise`	the precision of the inverse CDF of local min p-values. This will be used only if FINV is not specified. Defaults to 10000
`simT`	number of simulations in getting the inverse CDF of the local minimum p values

Value

This function generates a list of detected change-points and clustered CNVs for all samples.

newcp a list of vectors presenting detected change-points, which is in marker index units. Length of the list is the number of samples or sequences

h a list of vectors presenting the bandwidth used for this detected change-points. Length of the list is the number of samples

cnv.state state of detected CNV segments, duplication or deletion

cnv.start a list of vectors presenting the start position of CNV segments

cnv.end a list of vectors presenting the end position of CNV segments

See Also

multiSaRa for processing the screening and ranking steps for single sequence

---

`multiSaRa` *Screening procedure processing single sequence to find local maximizers of the local dignostic statistic*

Description

This function runs the screening step under multiple bandwidths processing a single sequence.

Usage

```
multiSaRa(
  Y,
  h1 = 3 * round(log10(length(Y))),
  h2 = 2 * round(log10(length(Y))),
  h3 = round(log10(length(Y))),
  FINV = FINV,
  precise = precise,
  sigma = sigma
)
```

Arguments

`Y`	the vector of the intensities of markers
`h1`	the bandwidth 1 for the screening procedure, defaults to 5
`h2`	the bandwidth 2 for the screening procedure, defaults to 10
`h3`	the bandwidth 3 for the screening procedure, defaults to 15
`FINV`	the inverse CDF of the local minimum p-values, approximated by function fInverse()
`precise`	the precision of the inverse CDF of local min p-values. This will be used only if FINV is not specified. Defaults to 10000
`sigma`	the standard deviation for the intensities between two adjacent change-points, defaults to NULL

Value

The return is a list of index with local minimum p values at each bandwidth.

See Also

SARA for processing the screening and ranking steps using single bandwidth

Index