Package ‘breakpoint’

November 9, 2014

Type Package

Title An R Package for Multiple Break-Point Detection via the Cross-Entropy Method

Version 1.1

Date 2014-11-08

Author Priyadarshana W.J.R.M. and Georgy Sofronov

Maintainer Priyadarshana W.J.R.M. <madawa.weerasinghe@mq.edu.au>

Description Implements the cross-entropy (CE) method, which is a model based stochastic optimization technique to estimate both the number and their corresponding locations of break-points in biological sequences of continuous and discrete measurements as described in Priyadarshana and Sofronov (2014, 2012a, 2012b).

License GPL(>=2)

Depends R (>= 2.5.0)

Imports ggplot2, MASS, msm, doMC, doSNOW, snow, foreach

Suggests parallel

R topics documented:

`breakpoint-package` *Multiple Break-Point Detection via the Cross-Entropy Method*

Description

The breakpoint package implements variants of the Cross-Entropy (CE) method proposed in Priyadarshana and Sofronov (2014, 2012a and 2012b) to estimate both the number and the corresponding locations of break-points in biological sequences of continuous and discrete measurements. The proposed method is primarily built to detect multiple break-points in genomic sequences. However, it can be easily extended and applied to other problems.

Details

Package:	breakpoint
Type:	Package
Version:	1.1
Date:	2014-11-08
License:	GPL 2.0

"breakpoint"" package provides estimates on both the number as well as the corresponding locations of break-points. The algorithms utilize the Cross-Entropy (CE) method, which is a model based stochastic optimization procedure to obtain the estimates on location. Model selection procedures based on penalized likelihood methods are used to obtain the number of break-points. In analyzing continuous data, it uses the modified BIC introduced by Zhang & Siegmund (2007). In discrete data analysis it uses the general BIC. Current implementation of the methodology works as an exact search method in estimating the number of break-points. A parallel implementation of the algorithm can be carried-out in Unix/Linux/MAC OS X and Windows operating systems with the use of "doMC", "parallel", "snow" and "doSNOW" packages.

Author(s)

Priyadarshana, W.J.R.M. and Sofronov, G.

Maintainer: Priyadarshana, W.J.R.M. <madawa.weerasinghe@mq.edu.au>

References

Priyadarshana, W. J. R. M., Sofronov G. (2014). Multiple Break-Points Detection in array CGH Data via the Cross-Entropy Method, IEEE/ACM Transactions on Computational Biology and Bioinformatics, no. 1, pp. 1, PrePrints, doi:10.1109/TCBB.2014.2361639, ISSN: 1545-5963.

Priyadarshana, W. J. R. M. and Sofronov, G. (2012a). A Modified Cross- Entropy Method for Detecting Multiple Change-Points in DNA Count Data. In Proc. of the IEEE Conference on Evolutionary Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.

Priyadarshana, W. J. R. M. and Sofronov, G. (2012b). The Cross-Entropy Method and Multiple Change-Points Detection in Zero-Inflated DNA read count data. In: Y. T. Gu, S. C. Saha (Eds.) The 4th International Conference on Computational Methods (ICCM2012), 1-8, ISBN 978-1-921897-54-2.

Rubinstein, R., and Kroese, D. (2004) The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New York.

Zhang, N.R., and Siegmund, D.O. (2007) A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. Biometrics, 63, 22-32.

`CE.NB` *Multiple Break-point Detection via the CE Method with Negative Binomial Distribution*

Description

Performs calculations to estimate both the number of break-points and their corresponding locations of discrete measurements with the CE method. Negative binomial distribution is used to model the over-dispersed discrete (count) data. This function supports the simulation of break-point locations in the CE algorithm based on either the four parameter beta distribution or truncated normal distribution. The general BIC is used to select the optimal number of break-points.

Usage

```
CE.NB(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8, b = 0.8,
distyp = 1, parallel = FALSE)
```

Arguments

`data` data to be analysed. A single column array or a data frame.

`Nmax` maximum number of break-points. Default value is 10.

`eps` the cut-off value for the stopping criterion in the CE method. Default value is 0.01.

`rho` the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.

`M` sample size to be used in simulating the locations of break-points. Default value is 200.

`h` minimum aberration width. Default is 5.

`a` a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.

`b` a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.

`distyp` distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.

`parallel` A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In Windows OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

The negative binomial (NB) distribution is used to model the discrete (count) data. NB model is preferred over the Poisson model when over-dispersion is observed in the count data. A performance function score (BIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points. The solution that minimizes the BIC with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations and the number of break-points are given in the console.

Value

A list is returned with following items:

`No.BPs` The number of break-points in the data that is estimated by the CE method

`BP.Loc` A vector of break-point locations.

Author(s)

Priyadarshana, W.J.R.M. <madawa.weerasinghe@mq.edu.au>

References

Priyadarshana, W. J. R. M. and Sofronov, G. (2012a) A Modified Cross- Entropy Method for Detecting Multiple Change-Points in DNA Count Data, In Proc. of the IEEE Conference on Evolutionary Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.

Priyadarshana, W. J. R. M. and Sofronov, G. (2012b) The Cross-Entropy Method and Multiple Change-Points Detection in Zero-Inflated DNA read count data, In: Y. T. Gu, S. C. Saha (Eds.) The 4th International Conference on Computational Methods (ICCM2012), 1-8, ISBN 978-1-921897-54-2.

Rubinstein, R., and Kroese, D. (2004) The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New York.

Schwarz, G. (1978) Estimating the dimension of a model, The Annals of Statistics, 6(2), 461-464.

See Also

`CE.ZINB` for CE with zero-inflated negative binomial, `profilePlot` to obtain mean profile plot.

Examples

```
#### Simulated data example ###
segs <- 6 # Number of segements
M <- c(1500, 2200, 800, 2500, 1000, 2000) # Segment width
#true.locations <- c(1501, 3701, 4501, 7001, 8001)  # True break-point locations
seg <- NULL
p <- c(0.45, 0.25, 0.4, 0.2, 0.3, 0.6) # Specification of ps for each segment
for(j in 1:segs){
  seg <- c(seg, rnbinom(M[j], size =10, prob = p[j]))
}
simdata <- as.data.frame(seg)
rm(p, M, seg, segs, j)
#plot(data[, 1])

## Not run:
## CE with the four parameter beta distribution ##

obj1 <- CE.NB(simdata, distyp = 1, parallel = TRUE) # Parallel computation
obj1

profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution ##

obj2 <- CE.NB(simdata, distyp = 2, parallel = TRUE) # Parallel computation
```

```
obj2

profilePlot(obj1, simdata) # To obtain the mean profile plot

## End(Not run)
```

CE.Normal *Multiple Break-point Detection via the CE Method for Continuous Data*

Description

This function performs calculations to estimate both the number of break-points and their corresponding locations of continuous measurements with the CE method. The normal distribution is used to model the observed continuous data. This function supports the simulation of break-point locations based on the four parameter beta distribution and truncated normal distribution. The modified BIC proposed by Zhang and Siegmund (2007) is used to select the optimal number of break-points.

Usage

```
CE.Normal(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8,
b = 0.8, distyp = 1, parallel = FALSE)
```

Arguments

`data` data to be analysed. A single column array or a data frame.

`Nmax` maximum number of break-points. Default value is 10.

`eps` the cut-off value for the stopping criterion in the CE method. Default value is 0.01.

`rho` the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.

`M` sample size to be used in simulating the locations of break-points. Default value is 200.

`h` minimum aberration width. Default is 5.

`a` a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.

`b` a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.

`distyp` distributions to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.

`parallel` A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In Windows OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

The normal distribution is used to model the continuous data. A performance function score (mBIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points. The solution that maximizes the mBIC with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations and the number of break-points are given in the console.

Value

A list is returned with following items:

`No.BPs`	The number of break-points in the data that is estimated by the CE method
`BP.Loc`	A vector of break-point locations.

Author(s)

Priyadarshana, W.J.R.M. <madawa.weerasinghe@mq.edu.au>

References

Priyadarshana, W. J. R. M., Sofronov G. (2014) Multiple Break-Points Detection in array CGH Data via the Cross-Entropy Method, IEEE/ACM Transactions on Computational Biology and Bioinformatics, no. 1, pp. 1, PrePrints, doi:10.1109/TCBB.2014.2361639, ISSN: 1545-5963.

Priyadarshana, W. J. R. M. and Sofronov, G. (2012) A Modified Cross- Entropy Method for Detecting Multiple Change-Points in DNA Count Data, In Proc. of the IEEE Conference on Evolutionary Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.

Rubinstein, R., and Kroese, D. (2004) The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New York.

Zhang, N.R., and Siegmund, D.O. (2007) A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. Biometrics, 63, 22-32.

See Also

`profilePlot` to obtain mean profile plot.

Examples

```
data(ch1.GM03563)
## Not run:
## CE with four parameter beta distribution ##
obj1 <- CE.Normal(ch1.GM03563, distyp = 1, parallel =TRUE)
profilePlot(obj1, simdata)

## CE with truncated normal distribution ##
obj2 <- CE.Normal(ch1.GM03563, distyp = 2, parallel =TRUE)
profilePlot(obj2, simdata)

## End(Not run)
```

`CE.ZINB` *Multiple Break-point Detection via the CE Method with Zero-Inflated Negative Binomial Distribution*

Description

Performs calculations to estimate both the number of break-points and their corresponding locations of discrete measurements with the CE method. Zero-inflated negative binomial distribution is used to model the excess zero observations and to model over-dispersion in the observed discrete (count) data. This function supports the simulation of break-point locations in the CE algorithm based on the four parameter beta distribution and truncated normal distribution. The general BIC is used to select the optimal number of break-points.

Usage

```
CE.ZINB(data, Nmax = 10, eps = 0.01, rho = 0.05, M = 200, h = 5, a = 0.8,
b = 0.8, distyp = 1, parallel = FALSE)
```

Arguments

`data` data to be analysed. A single column array or a data frame.

`Nmax` maximum number of break-points. Default value is 10.

`eps` the cut-off value for the stopping criterion in the CE method. Default value is 0.01.

`rho` the fraction which is used to obtain the best performing set of sample solutions (i.e., elite sample). Default value is 0.05.

`M` sample size to be used in simulating the locations of break-points. Default value is 200.

`h` minimum aberration width. Default is 5.

`a` a smoothing parameter value. It is used in the four parameter beta distribution to smooth both shape parameters. When simulating from the truncated normal distribution, this value is used to smooth the estimates of the mean values. Default is 0.8.

`b` a smoothing parameter value. It is used in the truncated normal distribution to smooth the estimates of the standard deviation. Default is 0.8.

`distyp` distribution to simulate break-point locations. Options: 1 = four parameter beta distribution, 2 = truncated normal distribution. Default is 1.

`parallel` A logical argument specifying if parallel computation should be carried-out (TRUE) or not (FALSE). By default it is set as ‘FALSE’. In Windows OS systems "snow" functionalities are used, whereas in Unix/Linux/MAC OSX "multicore" functionalities are used to carryout parallel computations with the maximum number of cores available.

Details

Zero-inflated negative binomial (ZINB) distribution is used to model the discrete (count) data. ZINB model is preferred over the NB model when both excess zero values and over-dispersion observed in the count data. A performance function score (BIC) is calculated for each of the solutions generated by the statistical distribution (four parameter beta distribution or truncated normal distribution), which is used to simulate break-points from no break-point to the user provided maximum number of break-points. The solution that minimizes the BIC with respect to the number of break-points is reported as the optimal solution. Finally, a list containing a vector of break-point locations and the number of break-points are given in the console.

Value

A list is returned with following items:

No.BPs	The number of break-points in the data that is estimated by the CE method
BP.Loc	A vector of break-point locations.

Author(s)

Priyadarshana, W.J.R.M. <madawa.weerasinghe@mq.edu.au>

References

Priyadarshana, W. J. R. M. and Sofronov, G. (2012a) A Modified Cross- Entropy Method for Detecting Multiple Change-Points in DNA Count Data, In Proc. of the IEEE Conference on Evolutionary Computation (CEC), 1020-1027, DOI: 10.1109/CEC.2012.6256470.

Priyadarshana, W. J. R. M. and Sofronov, G. (2012b) The Cross-Entropy Method and Multiple Change-Points Detection in Zero-Inflated DNA read count data, In: Y. T. Gu, S. C. Saha (Eds.) The 4th International Conference on Computational Methods (ICCM2012), 1-8, ISBN 978-1-921897-54-2.

Rubinstein, R., and Kroese, D. (2004) The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New York.

Schwarz, G. (1978) Estimating the dimension of a model, The Annals of Statistics, 6(2), 461-464.

See Also

CE.NB for CE with negative binomial, profilePlot to obtain mean profile plot.

Examples

```
#### Simulated data example ###
# gamlss R package is used to simulate data from the ZINB.

## Not run:
library(gamlss)
segs <- 6 # Number of segements
M <- c(1500, 2200, 800, 2500, 1000, 2000) # Segment width
#true.locations <- c(1501, 3701, 4501, 7001, 8001)  # True break-point locations
seg <- NULL
p <- c(0.6, 0.1, 0.3, 0.05, 0.2, 0.4) # Specification of ps on each segment
sigma.val <- c(1,2,3,4,5,6) # Specification of sigma vlaues

for(j in 1:segs){
  seg <- c(seg, rZINBI(M[j], mu = 300, sigma = sigma.val[j], nu = p[j]))
}

simdata <- as.data.frame(seg)
```

```
rm(p, M, seg, segs, j, sigma.val)
#plot(data[, 1])

## CE with the four parameter beta distribution ##

obj1 <- CE.ZINB(simdata, distyp = 1, parallel = TRUE) # Parallel computation
obj1

profilePlot(obj1, simdata) # To obtain the mean profile plot

## CE with truncated normal distribution ##

obj2 <- CE.ZINB(simdata, distyp = 2, parallel = TRUE) # Parallel computation
obj2

profilePlot(obj2, simdata) # To obtain the mean profile plot

## End(Not run)
```

`ch1.GM03563` *Fibroblast cell line (GM03563) data*

Description

Chromosome 1 of cell line GM03563

Usage

```
data("ch1.GM03563")
```

Format

A single column data frame with 135 observations that corresponds to chromosome 1 of cell line GM03563.

`log2ratio` normalized average of the log base 2 test over reference ratio data

Details

This data set is extracted from a single experiments on 15 fibroblast cell lines with each array containing over 2000 (mapped) BACs spotted in triplicate discussed in Snijders et al.(2001). Data corresponds to the chromosome 1 of cell line GM03563.

References

Snijders,A.M. et al. (2001) Assembly of microarrays for genome-wide measurement of DNA copy number. Nature Genetics, 29, 263-26.

Examples

```
data(ch1.GM03563)
## Not run:
## CE with four parameter beta distribution ##
obj1 <- CE.Normal(ch1.GM03563, distyp = 1, parallel =TRUE)
profilePlot(obj1, ch1.GM03563)

## CE with truncated normal distribution ##
obj2 <- CE.Normal(ch1.GM03563, distyp = 2, parallel =TRUE)
profilePlot(obj2, ch1.GM03563)

## End(Not run)
```

profilePlot *Mean profile plot*

Description

Plotting function to obtain mean profile plot of the data based on the estimates of the break-points through CE method. An R object created from the CE.Normal, CE.NB or CE.ZINB is required. User can alter the axes names.

Usage

```
profilePlot(obj, data, x.label = "Data Sequence", y.label = "Value")
```

Arguments

obj	R object created from CE.Normal, CE.NB or CE.ZINB.
data	data to be analysed. A single column array or a data frame.
x.label	x axis label. Default is "Data Sequence".
y.label	y axis label. Default is "Value".

Author(s)

Priyadarshana, W.J.R.M. <madawa.weerasinghe@mq.edu.au>

See Also

CE.Normal, CE.NB, CE.ZINB.

Examples

```
data(ch1.GM03563)
## Not run:
## CE with four parameter beta distribution ##
obj1 <- CE.Normal(ch1.GM03563, distyp = 1, parallel =TRUE)
profilePlot(obj1)

## CE with truncated normal distribution ##
obj2 <- CE.Normal(ch1.GM03563, distyp = 2, parallel =TRUE)
profilePlot(obj2)

## End(Not run)
```

Index