Package ‘Segmentor3IsBack’

Version 2.0
Date 2016-06-25
Title A Fast Segmentation Algorithm
Author Alice Cleynen, Guillaume Rigaill, Michel Koskas
Maintainer Alice Cleynen <alice.cleynen@umontpellier.fr>
Description Performs a fast exact segmentation on data and allows for use of various cost functions.
License GPL (>= 2)
Depends R (>= 2.0.0), methods
Repository CRAN
Repository/R-Forge/Project segmentor3isbac
Repository/R-Forge/Revision 31
Repository/R-Forge/DateTimeStamp 2016-06-22 13:18:27
Date/Publication 2016-06-30 09:17:38
NeedsCompilation yes

R topics documented:

Segmentor3IsBack-package ........................................ 2
BestSegmentation ................................................. 3
getBreaks .......................................................... 5
getBreaks-methods ............................................... 6
getCompression .................................................... 6
getCompression-methods ........................................ 7
getCost ............................................................. 7
getCost-methods .................................................. 8
getData ............................................................ 9
getData-methods .................................................. 10
getKmax ........................................................... 10
getKmax-methods ................................................ 11
getLikelihood ..................................................... 11
getLikelihood-methods .......................................... 12
Implementation of the Pruned Dynamic Programming Algorithm for the exact optimal segmentation of profiles

Description

Exact change-point algorithm for the segmentation of profiles according to the log-likelihood criterion for 5 possible models: Poisson, Gaussian homoscedastic, negative binomial, Gaussian with constant mean and Exponential.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>Segmentor3IsBack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.5</td>
</tr>
<tr>
<td>Date</td>
<td>2013-03-25</td>
</tr>
<tr>
<td>License</td>
<td>GPL</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>

Author(s)

Alice Cleynen

Maintainer: Alice Cleynen <alice.cleynen@agroparistech.fr>
References


Examples

N=1000
x=rbinom(5*N, size=1.3, prob=rep(c(0.7,0.2,0.01,0.2,0.8),each=N))
res=Segmentor(data=x, model=3, Kmax=20, keep=TRUE);
# Finds the optimal segmentation in up to 20 segments with respect to
# the negative binomial model, keeping cost matrix.
Cr<-SelectModel(res, penalty='oracle', keep=FALSE)
Cr
# chooses the number of segments in the segmentation of x, not keeping
# values of constants for slope heuristic.
Best<-BestSegmentation(res, K=Cr, t=2*N)
matplot(Best$bestCost, type='l', lty=2)
points(apply(Best$bestCost,2,which.min),apply(Best$bestCost,2,min),pch=20,col=1:(Cr-1))
apply(Best$bestCost, 2, which.min)
getBreaks(res)[Cr,1:(Cr-1)]
# computes and plots cost of best segmentation in Cr segments with
# change-point t, and compares result with change-point estimates.
Best$bestSeg
# returns the optimal segmentation in Cr segments with t as a
# change-point
Description

This function is used to compute the cost of the best segmentation in K segments given the position of a change-point, and to return the optimal segmentation having a change-point at location t.

Usage

BestSegmentation(x, K, t = numeric(), compress = TRUE)

Arguments

x
An object of class Segmentor returned by function Segmentor

K
The number of segments of the segmentation for which the cost or best segmentation is wanted

t
The position for which the best segmentation with t as change-point is wanted

compress
A boolean stating whether data should be compressed prior to segmentation

Value

bestCost
A matrix of size n*K: the cost of the optimal segmentation with jth change-point

bestSeg
If a t has been specified, a vector of size K+1 containing values of indicating the optimal segmentation with t as a change-point

Author(s)

Alice Cleynen, Michel Koskas and Guillem Rigaill

Maintainer: Who to complain to <alice.cleynen@agroparistech.fr>

References


Examples

require(Segmentor3IsBack);
N = 2000
x = rbinom(3 * N, size = 1.3, prob = rep(c(0.7, 0.2, 0.01), each = N));
res = Segmentor(data = x, model = 3, Kmax = 10, keep = TRUE);
# Finds the optimal segmentation in up to 10 segments with respect to
# the negative binomial model.
K = 3
Best <- BestSegmentation(res, K = 3, t = 3000, compress = FALSE)
matplot(Best$bestCost, type = 'l', lty = 2)
points(apply(Best$bestCost, 2, which.min), apply(Best$bestCost, 2, min), pch = 20, col = 1:(K + 1))
getBreaks

```r
apply(Best$bestCost, 2, which.min)
getBreaks(res)[K, 1:(K-1)]
# computes and plots cost of best segmentation in 3 segments with
# change-point t, and compares result with change-point estimates.
Best$bestSeg
# returns the optimal segmentation in 3 segments with 3000 as a
# change-point
Best<-BestSegmentation(res, K=3, t=3000, compress=TRUE)
Best$bestSeg
# this segmentation usually does not make sense because of the
# compress option. t has to be adapted consequently
```

---

**getBreaks**

**Generic function**

**Description**

Generic function

**Usage**

```r
getBreaks(object)
```

**Arguments**

- `object`: An object of class Segmentor

**Details**

Returns the slot breaks of an object of class Segmentor

**Value**

The ML estimates of the breakpoints of the optimal segmentation for each value of k

**Author(s)**

Alice Cleynen

**References**


getCompression

See Also

getLikelihood

Examples

x = new("Segmentor") # new Segmentor object
getBreaks(x) # retrieves the breaks slot from x

description

getBreaks-methods  ~~ Methods for Function getBreaks ~~

Methods

signature(object = "Segmentor") Retrieves the change-point location of segmentations

description

getCompression  Generic function

Description

Generic function

Usage

getCompression(object)

Arguments

object  An object of class Segmentor

Details

Returns the slot compression of an object of class Segmentor

Value

the value of compression that was used when analysing the data.

Author(s)

Alice Cleynen
References


See Also

gMeans

Examples

x=new("Segmentor") # new Segmentor object
getCompression(x) # retrieves the compression slot from x

gCompression-methods

Description

~ Methods for function getCompression ~

Methods

signature(object = "Segmentor") Retrieves used compression value from an object of class Segmentor

getCost

Generic function

Description

Generic function

Usage

gCost(object)

Arguments

object An object of class Segmentor
getCost-methods

Details

Returns the slot Cost of an object of class Segmentor

Value

The matrix of Cost of the segmentation in i segments up to point j

Author(s)

Alice Cleynen

References


See Also

getLikelihood; getBreaks

Examples

x=new("Segmentor") # new Segmentor object
gcost(x) # retrieves the Cost slot from x

---

gcost-methods ~ Methods for Function getCost ~

Description

~ Methods for function getCost ~

Methods

signature(object = "Segmentor") Retrieves the cost of segmenting signal up to j in i segments
getData

Generic function

Description
Generic function

Usage
getData(object)

Arguments

object
An object of class Segmentor

Details
Returns the slot data of an object of class Segmentor

Value
The vector of data to be segmented

Author(s)
Alice Cleynen

References


See Also
getData

Examples
x=new("Segmentor") # new Segmentor object
data(x) # retrieves the data slot from x
### Description

~~ Methods for function `getData` ~~

~~ Methods for function `getKmax` ~~

### Methods

`signature(object = "Segmentor")` Retrieves the data to be segmented

### `getKmax`

*Generic function*

#### Description

Generic function

#### Usage

`getKmax(object)`

#### Arguments

- `object` An object of class `Segmentor`

#### Details

Returns the slot Kmax of an object of class `Segmentor`

#### Value

An integer of the maximum number of segments considered for the analysis of the dataset.

#### Author(s)

Alice Cleynen

#### References


**getKmax-methods**

**See Also**

`getModel`

**Examples**

```r
x <- new("Segmentor") # new EBS object
getKmax(x) # retrieves the Kmax slot from x
```

---

**Description**

~~ Methods for function getKmax ~~

**Methods**

`signature(object = "Segmentor")` Retrieves maximum number of segments considered from an object of class Segmentor

---

**Description**

Generic function

**Usage**

`getLikelihood(object)`

**Arguments**

`object` An object of class Segmentor

**Details**

Returns the slot likelihood of an object of class Segmentor

**Value**

The opposite of the log-likelihood of the optimal segmentation for each value of \( k \)

**Author(s)**

Alice Cleynen
getMean

References


See Also

getBreaks

Examples

x=new("Segmentor") # new Segmentor object
getLikelihood(x) # retrieves the likelihood slot from x

ggLikelihood-methods  ~~ Methods for Function ggLikelihood ~~

Description

~~ Methods for function ggLikelihood ~~

Methods

signature(object = "Segmentor") Retrieves the likelihood of the segmentation from an object of class Segmentor

ggetMean

Generic function

Description

Generic function

Usage

ggetMean(object)

Arguments

object An object of class Segmentor

Details

Returns the slot Mean of an object of class Segmentor
Value

If model is Variance, the value of the mean used for the analysis.

Author(s)

Alice Cleynen

References


See Also

getoOverdispersion

Examples

```r
x=new("Segmentor", model="Variance", mean=2.2) # new Segmentor object
getMean(x) # retrieves the Mean slot from x
```

Description

~~ Methods for function getMean ~~

Methods

signature(object = "Segmentor") Retrieves estimate value of mean used from an object of class Segmentor
getModel

**Generic function**

**Description**

Generic function

**Usage**

```r
gemodel(object)
```

**Arguments**

- `object` An object of class Segmentor

**Details**

Returns the slot Model of an object of class Segmentor

**Value**

An object of class string returning the model used for the analysis of the dataset.

**Author(s)**

Alice Cleynen

**References**


**See Also**

`getKmax`

**Examples**

```r
x=new("Segmentor") # new Segmentor object
gemodel(x) # retrieves the Model slot from x
```
getModel-methods

Description

~~ Methods for function getModel ~~

Methods

signature(object = "Segmentor") Retrieves model used from an object of class Segmentor

getOverdispersion

Generic function

Description

Generic function

Usage

getOverdispersion(object)

Arguments

object An object of class Segmentor

Details

Returns the slot Overdispersion of an object of class Segmentor

Value

If model is Negative Binomial, the value of the overdispersion used for the analysis.

Author(s)

Alice Cleynen

References

See Also

getMean

Examples

x = new("Segmentor")  # new Segmentor object
getoOverdispersion(x)  # retrieves the overdispersion slot from x

description

getOverdispersion-methods

~~ Methods for Function getOverdispersion ~~

Description

~~ Methods for function getOverdispersion ~~

Methods

signature(object = "Segmentor")  Retrieves estimate value of overdispersion used from an object of class Segmentor

getParameters

Generic function

Description

Generic function

Usage

getParameters(object)

Arguments

object  An object of class Segmentor

Details

Returns the slot parameters of an object of class Segmentor

Value

The ML estimates of each segment’s parameter of the optimal segmentation for each value of k
getParameters-methods

Author(s)
Alice Cleynen

References

See Also
getBreaks

Examples
x=new("Segmentor") # new Segmentor object
getParameters(x) # retrieves the parameters slot from x

Description
Methods
signature(object = "Segmentor") Retrieves the estimates of the parameters of each segment of the segmentation from an object of class Segmentor

getPos

Description
Generic function

Usage
getPos(object)

Arguments
object An object of class Segmentor
getPos-methods

Details

Returns the slot Pos of an object of class Segmentor

Value

The matrix of the last change-point location of the segmentation in i segments up to point j

Author(s)

Alice Cleynen

References


See Also

getCost; getBreaks

Examples

x = new("Segmentor") # new Segmentor object
getPos(x) # retrieves the pos slot from x

getPos-methods

Description

~~ Methods for function getPos ~~

Methods

signature(object = "Segmentor") Retrieves the last change-point location of segmenting signal up to j in i segments
Description

The functions are used for change-point problems. Given a loss function (Poisson, Normal homoscedastic, Negative Binomial, Normal Heteroscedastic (with given constant mean) or Exponential), the function Segmentor estimates the optimal segmentation with respect to the log-likelihood. The Segmentor gives estimates of the breakpoint locations as well as the loss function parameter of each segment.

Usage

Segmentor(data=numeric(), model=1, Kmax = 15, phi = numeric(), m = numeric(), keep=FALSE, bounds=c(0,0), compress = TRUE)

Arguments

data | A vector of observations to be segmented. Must have no missing values.
model | Integer between 1 and 4 giving the modelisation of the observed data, 1: poisson (default), 2: normal-homoscedastic, 3: negative binomial, 4: normal-heteroscedastic or 5: exponential
Kmax | The maximum number of segments wanted for the data. The Segmentor will find all optimal segmentations in 1 to Kmax segments.
phi | Needed only for the Negative Binomial distribution: the value of the inverse of the overdispersion parameter. If the user does not enter a value, the package uses a modified version of Johnson and Kotz’s estimator where the mean is replaced by the median.
m | Needed only for Normal Heteroscedastic distribution: the value of the constant mean. If not entered, the function uses the empirical mean of the data.
keep | a boolean stating whether or not to keep Cost and Position matrices
bounds | optional lower and upper bounds on the parameter to segment: will fasten the pruning and hence the algorithm if some values are not allowed (e.g. non negative numbers for Poisson distribution)
compress | A boolean stating whether data should be compressed prior to segmentation

Details

Package: Segmentor3IsBack
Type: Package
Version: 1.5
Date: 2013-03-25
License: GPL (>= 2)
Value

data The vector of observations to be segmented.
model Emission distribution (Poisson, Normal Homoscedastic, Negative Binomial or Normal Heteroscedastic or exponential)
breaks Matrix of size Kmax*Kmax of estimated change-point locations for each optimal segmentation in 1 to Kmax segments.
parameters Matrix of size Kmax*Kmax which elements are the estimated parameters for each segment of the optimal segmentation. If model is Poisson or Normal, the parameter corresponds to the mean of the signal in each segment. If model is Negative binomial, the parameter corresponds to the success-probability of the signal in each segment. If model is normal heteroscedastic, the parameter is the variance assuming known mean.
likelihood Vector of size Kmax of resulting negative log-likelihood for each optimal segmentation.
Cost Matrix of size Kmax x n containing the cost of the segmentation of signal up to point j in i segments
Pos Matrix of size Kmax x n containing the last change-point location of the segmentation of signal up to point j in i segments
overdispersion only if model = Negative Binomial, the value of the inverse of overdispersion used for the segmentation
mean only if model = Normal Heteroscedastic, the value of the mean used for the segmentation
compression The value of the compression factor used (compression>=1)

Author(s)

Alice Cleynen, Michel Koskas and Guillem Rigaill

Maintainer: Who to complain to <alice.cleynen@agroparistech.fr>

References


Examples

```r
require(Segmentor3IsBack);
N=2000
x=c(rpois(N,2.0),rpois(2*N,2.2),rpois(N,1.9));
res=Segmentor(data=x,Kmax=3);
# Finds the optimal segmentation in 1, 2 and 3 segments with respect to
#the Poisson model.

y=c(rnbinom(N,prob=0.3,size=0.15),rnbinom(2*N,prob=0.1,size=0.15),
rnbinom(N,prob=0.6,size=0.15),compress=FALSE)
res2=Segmentor(y, model=3,Kmax=10);
#Finds the optimal segmentation in 1 to 10 segments with respect to
#the Negative Binomial model, without compression of data.
```

Description

A class for Segmentation objects.

Objects from the Class

Objects can be created by calls of the form `new("Segmentor", ...)`. `new("Segmentor", ...)`: creates a new object with class Segmentor

Slots

data: Object of class "numeric", the vector of data
model: Object of class "character", the assumed distribution of the data
Kmax: Object of class "numeric", the maximum number of segments considered for the segmentation
breaks: Object of class "matrix", the breakpoints of the optimal segmentation for each value of k
mean: Object of class "numeric", if model = Variance, the mean used in the analysis
overdispersion: Object of class "numeric", if model = Negative Binomial, the overdispersion used in the analysis
parameters: Object of class "matrix", the ML estimates of the parameter of each segment of the optimal segmentation for each value of k
likelihood: Object of class "matrix", the likelihood of the segmentation for each value of k
Cost: Object of class "matrix", the cost of the segmentation up to point j in i segments
Pos: Object of class "matrix", the last change-point location of the segmentation up to point j in i segments
compression: Object of class "numeric", the value of the compression obtained prior to segmentation of the data
Methods

- **getData** signature(object = "Segmentor") retrieves data slot
- **getModel** signature(object = "Segmentor") retrieves model slot
- **getKmax** signature(object = "Segmentor") retrieves Kmax slot
- **getBreaks** signature(object = "Segmentor") retrieves breaks slot
- **getMean** signature(object = "Segmentor") retrieves mean slot
- **getOverdispersion** signature(object = "Segmentor") retrieves overdispersion slot
- **getLikelihood** signature(object = "Segmentor") retrieves likelihood slot
- **getCost** signature(object = "Segmentor") retrieves Cost slot
- **getParameters** signature(object = "Segmentor") retrieves parameters slot
- **getPos** signature(object = "Segmentor") retrieves Pos slot
- **getCompression** signature(object = "Segmentor") retrieves compression slot

Author(s)

Alice Cleynen

See Also

SelectModel

Description

This function is used to select the number of segments in the segmentation of the data according to several criterion including BIC, AIC, mBIC and oracle penalties.

Usage

```
SelectModel(x, penalty="oracle", seuil=n/log(n), keep=FALSE, greatjump=FALSE)
```

Arguments

- **x** An object of class Segmentor returned by function Segmentor
- **penalty** An object of class string stating which penalty to use in the model selection criteria. Accepted penalties are BIC, AIC, Zhang’s modified BIC: mBIC and oracle penalties: 'oracle'. In the case of Normal segmentation, criterion developed by Lebarbier, in the case of Poisson and Negative Binomial segmentation, criterion developed by Cleynen and Lebarbier. Default is oracle
- **seuil** If penalty='oracle', an Integer for the threshold to use for the slope heuristic. Default value is n/log(n)
keep  A Boolean stating whether or not to keep the values of the criterion. Default is FALSE.
greatjump  If penalty is "oracle", a boolean stating whether to use the greatest jump (TRUE) or the threshold for the slope heuristic. Default is FALSE.

Details

Package: Segmentor3IsBack
Type: Package
Version: 1.5
Date: 2013-03-25
License: GPL (>= 2)

Value

k  The number of segments selected.
crit  If keep=TRUE, a vector of criterion value for each possible K.

Author(s)

Alice Cleynen, Michel Koskas and Guillel Rigaill

Maintainer: Who to complain to <alice.cleynen@agroparistech.fr>

References


Examples

```
require(Segmentor3IsBack);
N=2000
x=rbinom(5*N, size=1.3, prob=rep(c(0.7,0.2,0.01,0.2,0.8),each=N))
res=Segmentor(data=x,model=3,Kmax=20);
# Finds the optimal segmentation in up to 20 segments with respect to
# the negative binomial model.
Cr<-SelectModel(res,penalty='oracle',keep=FALSE)
Cr
#chooses the number of segments in the segmentation of x using
# an oracle-inequality approach

N=250
x=rpois(10*N, rep(c(8,1,5,3,16,33,2,12,7,1),each=N))
res=Segmentor(data=x,model=3,Kmax=40);
# Finds the optimal segmentation in up to 40 segments with respect to
#the poisson model.
Cr<-SelectModel(res,penalty='BIC',keep=FALSE)
Cr
#chooses the number of segments in the segmentation of x using
# the BIC approach
```

---

**show-methods**

~~ Methods for Function `show` in Package `methods` ~~

**Description**

~~ Methods for function `show` in package `methods` ~~

**Methods**

```
signature(object = "ANY") Generic function, see description using ?show
signature(object = "Segmentor") Prints a summary of the Segmentor object
```
Index

*Topic **Breakpoints**
  getBreaks, 5
*Topic **Breaks**
  getBreaks-methods, 6
*Topic **Compression**
  getCompression, 6
*Topic **Cost**
  getCost, 7
  getCost-methods, 8
*Topic **Data**
  getData, 9
*Topic **Kmax**
  getKmax, 10
  getKmax-methods, 11
*Topic **Likelihood**
  getLikelihood, 11
*Topic **Mean**
  getMean, 12
  getMean-methods, 13
*Topic **Model**
  getModel, 14
  getModel-methods, 15
*Topic **Overdispersion**
  getOverdispersion, 15
*Topic **PDPA**
  BestSegmentation, 3
  Segmentor, 19
  SelectModel, 22
*Topic **Position**
  getPos, 17
*Topic **Pos**
  getPos-methods, 18
*Topic **R-package**
  BestSegmentation, 3
  Segmentor, 19
  SelectModel, 22
*Topic **Segmentation**
  BestSegmentation, 3
  Segmentor, 19
  SelectModel, 22
*Topic **Segmentor**
  Segmentor-class, 21
  show-methods, 24
*Topic **changepoints**
  getBreaks, 5
  getCost, 7
  getData, 9
  getLikelihood, 11
  getParameters, 16
  getPos, 17
*Topic **changepoint**
  Segmentor3IsBack-package, 2
*Topic **classes**
  Segmentor-class, 21
*Topic **compression**
  getCompression-methods, 7
*Topic **counts**
  Segmentor3IsBack-package, 2
*Topic **data**
  getData-methods, 10
*Topic **exact**
  Segmentor3IsBack-package, 2
*Topic **likelihood**
  getLikelihood-methods, 12
*Topic **methods**
  getBreaks, 5
  getBreaks-methods, 6
  getCompression, 6
  getCompression-methods, 7
  getCost, 7
  getCost-methods, 8
  getData, 9
  getData-methods, 10
  getKmax, 10
  getKmax-methods, 11
  getLikelihood, 11
  getLikelihood-methods, 12
  getMean, 12
getMean-methods, 13
getModel, 14
getModel-methods, 15
getOverdispersion, 15
getOverdispersion-methods, 16
getParameters, 16
getParameters-methods, 17
getPos, 17
getPos-methods, 18
show-methods, 24

*Topic overdispersion
  getOverdispersion-methods, 16
*Topic parameters
  getParameters, 16
  getParameters-methods, 17
*Topic segmentation
  Segmentor3IsBack-package, 2
*Topic selection
  Segmentor3IsBack-package, 2
*Topic show
  show-methods, 24

BestSegmentation, 3
getBreaks, 5, 8, 12, 17, 18
getBreaks, Segmentor-method
  (getBreaks-methods), 6
getBreaks-methods, 6
getCompression, 6
getCompression, Segmentor-method
  (getCompression-methods), 7
getCompression-methods, 7
getCost, 7, 18
getCost, Segmentor-method
  (getCost-methods), 8
getCost-methods, 8
data, 9
data, Segmentor-method
  (getData-methods), 10
data-methods, 10
getKmax, 10, 14
getKmax, Segmentor-method
  (getKmax-methods), 11
getKmax-methods, 11
getLikelihood, 6, 8, 11
getLikelihood, Segmentor-method
  (getLikelihood-methods), 12
getLikelihood-methods, 12
getMean, 7, 12, 16
getMean, Segmentor-method
  (getMean-methods), 13
goal, 13
getModel, 9, 11, 14
getModel, Segmentor-method
  (getModel-methods), 15
goal, 15
getOverdispersion, 13, 15
getOverdispersion, Segmentor-method
  (getOverdispersion-methods), 16
goal, 16
getParameters, 16
getParameters, Segmentor-method
  (getParameters-methods), 17
goal, 17
goal, 17
getPos, Segmentor-method
  (getPos-methods), 18
goal, 18

print.Segmentor (Segmentor), 19
Segmentor, 19
Segmentor-class, 21
Segmentor-method, show
  (Segmentor-class), 21
Segmentor-method, getBreaks
  (Segmentor-class), 21
Segmentor-method, getCompression
  (Segmentor-class), 21
Segmentor-method, getCost
  (Segmentor-class), 21
Segmentor-method, getData
  (Segmentor-class), 21
Segmentor-method, getKmax
  (Segmentor-class), 21
Segmentor-method, getLikelihood
  (Segmentor-class), 21
Segmentor-method, getMean
  (Segmentor-class), 21
Segmentor-method, getModel
  (Segmentor-class), 21
Segmentor-method, getOverdispersion
  (Segmentor-class), 21
Segmentor-method, getParameters
  (Segmentor-class), 21
Segmentor-method, getPos
  (Segmentor-class), 21
INDEX

Segmentor3IsBack
(Se"gmentor3IsBack-package), 2
Segmentor3IsBack-package, 2
SelectModel, 22, 22
show, ANY-method (show-methods), 24
show, Segmentor-method (show-methods), 24
show-methods, 24