

# Package ‘cpm’

July 28, 2015

**Title** Sequential and Batch Change Detection Using Parametric and Nonparametric Methods

**Version** 2.2

**Date** 2015-07-09

**Depends** R (>= 2.15.0), methods

**Author** Gordon J. Ross

**Maintainer** Gordon J. Ross <gordon@gordonjross.co.uk>

**Description** Sequential and batch change detection for univariate data streams, using the change point model framework. Functions are provided to allow nonparametric distribution-free change detection in the mean, variance, or general distribution of a given sequence of observations. Parametric change detection methods are also provided for Gaussian, Bernoulli and Exponential sequences. Both the batch (Phase I) and sequential (Phase II) settings are supported, and the sequences may contain either a single or multiple change points.

**License** GPL-3

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-07-28 00:23:54

## R topics documented:

cpm-package	2
changeDetected	4
cpmReset	5
detectChangePoint	6
detectChangePointBatch	9
ForexData	12
getBatchThreshold	13
getStatistics	15
makeChangePointModel	16
processObservation	19
processStream	20
<b>Index</b>	<b>24</b>

## Description

An implementation of several different change point models (CPMs) for performing both parametric and nonparametric change detection on univariate data streams.

## Details

The CPM framework is an approach to sequential change detection (also known as Phase II process monitoring) which allows standard statistical hypothesis tests to be deployed sequentially. The main two general purpose functions in the package are `detectChangePoint` and `processStream` for detecting single and multiple change points respectively. The remainder of the functions allow for more precise control over the change detection procedure. To cite this R package in a research paper, please use `citation('cpm')` to obtain the reference, and BibTeX entry.

Note: this package has a manual titled "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk), which contains a full description of all the functions and algorithms in the package, as well as detailed instructions on how to use it.

If you would like to cite this package, the citation information is "G. J. Ross - Parametric and Non-parametric Sequential Change Detection in R: The cpm Package, Journal of Statistical Software, 2015, 66(3), 1-20"

### A Brief CPM Overview

Given a sequence  $X_1, \dots, X_n$  of random variables, the CPM works by evaluating a two-sample test statistic at every possible split point. Let  $D_{k,n}$  be the value of the test statistic when the sequence is split into the two samples  $\{X_1, X_2, \dots, X_k\}$  and  $\{X_{k+1}, X_{k+2}, \dots, X_n\}$ , and define  $D_n$  to be the maximum of these values.  $D_n$  is then compared to some threshold, with a change being detected if the threshold is exceeded.

In the sequential context, the observations are processed one-by-one, with  $D_t$  being computed based on the first  $t$  observations,  $D_{t+1}$  being computed based on the first  $t+1$  observations, and so on. The change detection time is defined as the first value of  $t$  where the threshold is exceeded. Supposing this occurs at time  $t = T$ , then the best estimate of the location of the change point is the value of  $k$  which maximised  $D_{k,T}$ . Writing  $\hat{\tau}$  for this, we have that  $\hat{\tau} \leq T$ .

The thresholds are chosen so that there is a constant probability of a false positive occurring after each observation. This leads to control of the Average Run Length ( $ARL_0$ ), defined as the expected number of observations received before a change is falsely detecting, assuming that no change has occurred.

The choice of test statistic in the CPM defines the class of changes which it is optimised towards detecting. This package implements CPMs using the following statistics. More details can be found in the references section:

- *Student*: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.
- *Bartlett*: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.

- **GLR**: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.
- **Exponential**: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.
- **GLRAdjusted** and **ExponentialAdjusted**: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.
- **FET**: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.
- **Mann-Whitney**: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.
- **Mood**: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.
- **Lepage**: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.
- **Kolmogorov-Smirnov**: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.
- **Cramer-von-Mises**: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.

For a fuller overview of the package which includes a description of the CPM framework and examples of how to use the various functions, please consult the full package manual titled "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package"

#### Author(s)

Gordon J. Ross <gordon@gordonjross.co.uk>

#### References

- Hawkins, D. , Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31
- Hawkins, D. , Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173
- Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.
- Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)
- Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116
- Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)
- Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030

Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

---

changeDetected                      *Tests Whether a CPM S4 Object Has Encountered a Change Point*

---

### Description

Tests whether an existing Change Point Model (CPM) S4 object has encountered a change point. It returns TRUE if a change has been encountered, otherwise FALSE.

Note that this function is part of the S4 object section of the cpm package, which allows for more precise control over the change detection process. For many simple change detection applications this extra complexity will not be required, and the detectChangePoint and processStream functions should be used instead.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

### Usage

```
changeDetected(cpm)
```

### Arguments

cpm                      The CPM S4 object which is to be tested for whether a change has occurred.

### Value

TRUE if a change has been detected, otherwise FALSE.

### Author(s)

Gordon J. Ross <[gordon@gordonjross.co.uk](mailto:gordon@gordonjross.co.uk)>

### See Also

[makeChangePointModel](#), [processObservation](#).

### Examples

```
#generate a sequence containing a single change point
x <- c(rnorm(100,0,1),rnorm(100,1,1))

#use a Student CPM
cpm <- makeChangePointModel(cpmType="Student", ARL0=500)

for (i in 1:length(x)) {
```

```
#process each observation in turn
cpm <- processObservation(cpm,x[i])

if (changeDetected(cpm)) {
  print(sprintf("change detected at observation %s",i))
  break
}
}
```

---

cpmReset

*Resets a CPM S4 Object*

---

### Description

Resets an existing Change Point Model (CPM) S4 object. Typically it will be used in situations where the CPM is processing a stream, and has encountered a change point. If the stream may contain multiple change points, the typical reaction procedure after detecting a change is to reset the CPM. This involves clearing its memory of all the observations prior to the change point, so that monitoring can be resumed starting with the next observation after the change point.

Note that this function is part of the S4 object section of the cpm package, which allows for more precise control over the change detection process. For many simple change detection applications this extra complexity will not be required, and the `detectChangePoint` and `processStream` functions should be used instead.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

### Usage

```
cpmReset(cpm)
```

### Arguments

cpm                    The CPM S4 object which is to be reset.

### Value

The reinitialised CPM.

### Author(s)

Gordon J. Ross <[gordon@gordonjross.co.uk](mailto:gordon@gordonjross.co.uk)>

### See Also

[makeChangePointModel](#), [processObservation](#), [changeDetected](#).

**Examples**

```

#generate a sequence containing two change points
x <- c(rnorm(200,0,1),rnorm(200,1,1),rnorm(200,0,1))

#vectors to hold the result
detectiontimes <- numeric()
changepoints <- numeric()

#use a Lepage CPM
cpm <- makeChangePointModel(cpmType="Lepage", ARL0=500)

i <- 0
while (i < length(x)) {
  i <- i + 1
  #process each observation in turn
  cpm <- processObservation(cpm,x[i])

  #if a change has been found, log it, and reset the CPM
  if (changeDetected(cpm) == TRUE) {
    print(sprintf("Change detected at observation %d", i))
    detectiontimes <- c(detectiontimes,i)

    #the change point estimate is the maximum D_kt statistic
    Ds <- getStatistics(cpm)
    tau <- which.max(Ds)

    if (length(changepoints) > 0) {
      tau <- tau + changepoints[length(changepoints)]
    }
    changepoints <- c(changepoints,tau)

    #reset the CPM
    cpm <- cpmReset(cpm)

    #resume monitoring from the observation following the
    #change point
    i <- tau
  }
}

```

---

detectChangePoint

*Detect a Single Change Point in a Sequence*


---

**Description**

This function is used to detect a single change point in a sequence of observations using the Change Point Model (CPM) framework for sequential (Phase II) change detection. The observations are processed in order, starting with the first, and a decision is made after each observation whether

a change point has occurred. If a change point is detected, the function returns with no further observations being processed. A full description of the CPM framework can be found in the papers cited in the reference section.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

## Usage

```
detectChangePoint(x, cpmType, ARL0=500, startup=20, lambda=NA)
```

## Arguments

- |         |   |
|---------|---|
| x       | A vector containing the univariate data stream to be processed.   |
| cpmType | <p>The type of CPM which is used. With the exception of the FET, these CPMs are all implemented in their two sided forms, and are able to detect both increases and decreases in the parameters monitored. Possible arguments are:</p> <ul style="list-style-type: none"> <li>• Student: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.</li> <li>• Bartlett: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.</li> <li>• GLR: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.</li> <li>• Exponential: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.</li> <li>• GLRAdjusted and ExponentialAdjusted: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.</li> <li>• FET: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.</li> <li>• Mann-Whitney: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Mood: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Lepage: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Kolmogorov-Smirnov: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Cramer-von-Mises: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> </ul> |

ARL <sub>0</sub>	Determines the $ARL_0$ which the CPM should have, which corresponds to the average number of observations before a false positive occurs, assuming that the sequence does not undergo a change. Because the thresholds of the CPM are computationally expensive to estimate, the package contains pre-computed values of the thresholds corresponding to several common values of the $ARL_0$ . This means that only certain values for the $ARL_0$ are allowed. Specifically, the $ARL_0$ must have one of the following values: 370, 500, 600, 700, ..., 1000, 2000, 3000, ..., 10000, 20000, ..., 50000.
startup	The number of observations after which monitoring begins. No change points will be flagged during this startup period. This must be set to at least 20.
lambda	A smoothing parameter which is used to reduce the discreteness of the test statistic when using the FET CPM. See [Ross and Adams, 2012b] in the References section for more details on how this parameter is used. Currently the package only contains sequences of ARL <sub>0</sub> thresholds corresponding to lambda=0.1 and lambda=0.3, so using other values will result in an error. If no value is specified, the default value will be 0.1.

**Value**

x	The sequence of observations which was processed.
changeDetected	TRUE if any $D_t$ exceeds the value of $h_t$ associated with the chosen $ARL_0$ , otherwise FALSE.
detectionTime	The observation after which the change point was detected, defined as the first observation after which $D_t$ exceeded the test threshold. If no change is detected, this will be equal to 0.
changePoint	The best estimate of the change point location. If the change is detected after the $t^{th}$ observation, then the change estimate is the value of $k$ which maximises $D_{k,t}$ . If no change is detected, this will be equal to 0.
Ds	The sequence of maximised $D_t$ statistics, starting from the first observation until the observation after which the change point was detected

**Author(s)**

Gordon J. Ross <gordon@gordonjross.co.uk>

**References**

- Hawkins, D., Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31
- Hawkins, D., Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173
- Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.
- Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)



Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116

Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)

Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030

Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

### See Also

[processStream](#), [detectChangePointBatch](#).

### Examples

```
## Use a Student-t CPM to detect a mean shift in a stream of Gaussian
## random variables which occurs after the 100th observation
x <- c(rnorm(100,0,1),rnorm(1000,1,1))
detectChangePoint(x,"Student",ARL0=500,startup=20)
```

```
## Use a Mood CPM to detect a scale shift in a stream of Student-t
## random variables which occurs after the 100th observation
x <- c(rt(100,5),rt(100,5)*2)
detectChangePoint(x,"Mood",ARL0=500,startup=20)
```

```
## Use a FET CPM to detect a parameter shift in a stream of Bernoulli
## observations. In this case, the lambda parameter acts to reduce the
## discreteness of the test statistic.
x <- c(rbinom(100,1,0.2), rbinom(1000,1,0.5))
detectChangePoint(x,"FET",ARL0=500,startup=20,lambda=0.3)
```

---

detectChangePointBatch

*Detect a Single Change Point in a Sequence*

---

### Description

This function is used to detect a single change point in a sequence of observations using the Change Point Model (CPM) framework for batch (Phase I) change detection. The observations are processed in one batch and information is returned regarding whether the sequence contains a change point. A full description of the CPM framework can be found in the papers cited in the reference section.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

**Usage**

```
detectChangePointBatch(x, cpmType, alpha=0.05, lambda=NA)
```

**Arguments**

x	A vector containing the univariate data stream to be processed.
cpmType	<p>The type of CPM which is used. With the exception of the FET, these CPMs are all implemented in their two sided forms, and are able to detect both increases and decreases in the parameters monitored. Possible arguments are:</p> <ul style="list-style-type: none"> <li>• Student: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.</li> <li>• Bartlett: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.</li> <li>• GLR: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.</li> <li>• Exponential: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.</li> <li>• GLRAdjusted and ExponentialAdjusted: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.</li> <li>• FET: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.</li> <li>• Mann-Whitney: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Mood: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Lepage: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Kolmogorov-Smirnov: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• Cramer-von-Mises: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> </ul>
alpha	the null hypothesis of no change is rejected if $D_n > h_n$ where n is the length of the sequence and $h_n$ is the upper alpha percentile of the test statistic distribution. Because computing the values of $h_n$ associated with each value of alpha is a laborious task, the package includes a function <code>getBatchThreshold</code> which returns the threshold associated with a particular choice of alpha and n. This function is called automatically whenever <code>detectChangePointBatch</code> is called, so the user need only provide the desired value of alpha. The allowable values

for this argument are 0.05, 0.01, 0.005, 0.001. If a different value is required then the user will need to compute it manually.

lambda A smoothing parameter which is used to reduce the discreteness of the test statistic when using the FET CPM. See [Ross and Adams, 2012b] in the References section for more details on how this parameter is used. Currently the package only contains sequences of ARL0 thresholds corresponding to lambda=0.1 and lambda=0.3, so using other values will result in an error. If no value is specified, the default value will be 0.1.

### Value

x The sequence of observations which was processed.

changeDetected TRUE if  $D_n$  exceeds the value of  $h_n$  associated with alpha, otherwise FALSE.

changePoint assuming a change was detected, this stores the most likely location of the change point, defined as the value of k which maximized  $D_k t$ . If no change is detected, this will be equal to 0.

threshold The value of  $h_n$  which corresponds to the specified alpha.

Ds The sequence of  $D_k t$  test statistics.

### Author(s)

Gordon J. Ross <gordon@gordonjross.co.uk>

### References

- Hawkins, D. , Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31
- Hawkins, D. , Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173
- Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.
- Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)
- Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116
- Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)
- Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030
- Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

### See Also

[detectChangePoint](#).

## Examples

```
## Use a Student-t CPM to detect a mean shift in a stream of Gaussian
## random variables which occurs after the 100th observation
x <- c(rnorm(100,0,1),rnorm(1000,1,1))
detectChangePointBatch(x,"Student",alpha=0.05)

## Use a Mood CPM to detect a scale shift in a stream of Student-t
## random variables which occurs after the 100th observation
x <- c(rt(100,5),rt(1000,5*2))
detectChangePointBatch(x,"Mood",alpha=0.05)

## Use a FET CPM to detect a parameter shift in a stream of Bernoulli
## observations. In this case, the lambda parameter acts to reduce the
## discreteness of the test statistic.
x <- c(rbinom(100,1,0.2), rbinom(1000,1,0.5))
detectChangePointBatch(x,"FET",alpha=0.05,lambda=0.3)
```

---

ForexData

*Foreign Exchange Data*

---

## Description

This dataset consists of a historical sequence of the exchange rates between the Swiss Franc (CHF) and the British Pound (GBP). The value of the exchange rate was recorded at three hour intervals running from October 21st 2002, to May 15th 2007. In total, 9273 observations were made.

For an example of how to use the `cpm` package to detect changes in this data set, please see the "Streams with multiple change points" section of the package vignette, available by typing: `vignette("cpm")`

## Usage

```
data(ForexData)
```

## Format

A three column matrix. Each observation is one row. The first column is the data, the second is the time of day at which the observation was recorded (1400 = 14:00pm, and so on), and the third is observed exchange rate.

---

getBatchThreshold      *Returns the Threshold Associated with a Type I Error Probability .*

---

### Description

When performing Phase I analysis within the CPM framework for a sequence of length  $n$ , the null hypothesis of no change is rejected if  $D_n > h_n$  for some threshold  $h_n$ . Typically this threshold is chosen to be the upper alpha quantile of the distribution of  $D_n$  under the null hypothesis of no change. Given a particular choice of alpha and  $n$ , this function returns the associated  $h_n$  threshold. Because these thresholds are laborious to compute, the package contains pre-computed values of  $h_n$  for alpha = 0.05, 0.01, 0.005 and 0.001, and for  $n < 10000$ .

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

### Usage

```
getBatchThreshold(cpmType, alpha, n, lambda=0.3)
```

### Arguments

cpmType

The type of CPM which is used. Possible arguments are:

- **Student**: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.
- **Bartlett**: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.
- **GLR**: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.
- **Exponential**: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.
- **GLRAdjusted** and **ExponentialAdjusted**: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.
- **FET**: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.
- **Mann-Whitney**: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.
- **Mood**: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.
- **Lepage**: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.

- Kolmogorov-Smirnov: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.
- Cramer-von-Mises: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.

alpha	the null hypothesis of no change is rejected if $D_n > h_n$ where $n$ is the length of the sequence and $h_n$ is the upper alpha percentile of the test statistic distribution.
n	the sequence length the value should be calculated for, i.e. the value of $n$ in $D_n$ .
lambda	A smoothing parameter which is used to reduce the discreteness of the test statistic when using the FET CPM. See [Ross and Adams, 2012b] in the References section for more details on how this parameter is used. Currently the package only contains sequences of ARL0 thresholds corresponding to lambda=0.1 and lambda=0.3, so using other values will result in an error. If no value is specified, the default value will be 0.1.

**Author(s)**

Gordon J. Ross <gordon@gordonjross.co.uk>

**References**

- Hawkins, D. , Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31
- Hawkins, D. , Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173
- Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.
- Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)
- Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116
- Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)
- Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030
- Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

**See Also**

[detectChangePointBatch](#).

**Examples**

```
## Returns the threshold for n=1000, alpha=0.05 and the Mann-Whitney CPM
h <- getBatchThreshold("Mann-Whitney", 0.05, 1000)
```

---

getStatistics	Returns the Test Statistics Associated With A CPM S4 Object
---------------	---

---

### Description

Returns the  $D_{k,t}$  statistics associated with an existing Change Point Model (CPM) S4 object. These statistics depend on the state of the object, which depends on the observations which have been processed to date. Calling this function returns the most recent set of statistics, which were generated after the previous observation was processed.

Note that this function is part of the S4 object section of the `cpm` package, which allows for more precise control over the change detection process. For many simple change detection applications this extra complexity will not be required, and the `detectChangePoint` and `processStream` functions should be used instead.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The `cpm` Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

### Usage

```
getStatistics(cpm)
```

### Arguments

`cpm`                    The CPM S4 object for which the test statistics are to be returned.

### Value

A vector containing the  $D_{k,t}$  statistics generated after the previous observation was processed.

### Author(s)

Gordon J. Ross <[gordon@gordonjross.co.uk](mailto:gordon@gordonjross.co.uk)>

### See Also

[makeChangePointModel](#), [processObservation](#), [changeDetected](#).

### Examples

```
#generate a sequence containing two change points
x <- c(rnorm(200,0,1),rnorm(200,1,1),rnorm(200,0,1))

#vectors to hold the result
detectiontimes <- numeric()
changepts <- numeric()

#use a Lepage CPM
```

```

cpm <- makeChangePointModel(cpmType="Lepage", ARL0=500)

i <- 0
while (i < length(x)) {
  i <- i + 1
  #process each observation in turn
  cpm <- processObservation(cpm,x[i])

  #if a change has been found, log it, and reset the CPM
  if (changeDetected(cpm) == TRUE) {
    print(sprintf("Change detected at observation %d", i))
    detectiontimes <- c(detectiontimes,i)

    #the change point estimate is the maximum D_kt statistic
    Ds <- getStatistics(cpm)
    tau <- which.max(Ds)

    if (length(changepoints) > 0) {
      tau <- tau + changepoints[length(changepoints)]
    }
    changepoints <- c(changepoints,tau)

    #reset the CPM
    cpm <- cpmReset(cpm)

    #resume monitoring from the observation following the
    #change point
    i <- tau
  }
}

```

---

makeChangePointModel *Creates a CPM S4 Object*

---

## Description

This function is used to create a change point model (CPM) S4 object. The CPM object can be used to process a sequence of data, one observation at a time. The CPM object maintains its state between each observation, and can be queried to obtain the  $D_{\{k, t\}}$  statistics, and to test whether a change has been detected.

Note that this function is part of the S4 object section of the `cpm` package, which allows for more precise control over the change detection process. For many simple change detection applications this extra complexity will not be required, and the `detectChangePoint` and `processStream` functions should be used instead.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The `cpm` Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)



**Usage**

```
makeChangePointModel(cpmType, ARL0=500, startup=20, lambda=NA)
```

**Arguments**

cpmType	<p>The type of CPM which is to be created. Possible arguments are:</p> <ul style="list-style-type: none"> <li>• <b>Student</b>: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.</li> <li>• <b>Bartlett</b>: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.</li> <li>• <b>GLR</b>: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.</li> <li>• <b>Exponential</b>: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.</li> <li>• <b>GLRAdjusted</b> and <b>ExponentialAdjusted</b>: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.</li> <li>• <b>FET</b>: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.</li> <li>• <b>Mann-Whitney</b>: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Mood</b>: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Lepage</b>: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Kolmogorov-Smirnov</b>: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Cramer-von-Mises</b>: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> </ul>
ARL0	<p>Determines the <math>ARL_0</math> which the CPM should have, which corresponds to the average number of observations before a false positive occurs, assuming that the sequence does not undergo a change. Because the thresholds of the CPM are computationally expensive to estimate, the package contains pre-computed values of the thresholds corresponding to several common values of the <math>ARL_0</math>. This means that only certain values for the <math>ARL_0</math> are allowed. Specifically, the <math>ARL_0</math> must have one of the following values: 370, 500, 600, 700, ..., 1000, 2000, 3000, ..., 10000, 20000, ..., 50000.</p>
startup	<p>The number of observations after which monitoring begins. No change points will be flagged during this startup period. This must be set to at least 20.</p>

**lambda** A smoothing parameter which is used to reduce the discreteness of the test statistic when using the FET CPM. See [Ross and Adams, 2012b] in the References section for more details on how this parameter is used. Currently the package only contains sequences of ARL0 thresholds corresponding to  $\lambda=0.1$  and  $\lambda=0.3$ , so using other values will result in an error. If no value is specified, the default value will be 0.1.

### Value

A CPM S4 object. The class of this object will depend on the value which has been passed as the `cpmType` argument.

### Author(s)

Gordon J. Ross <gordon@gordonjross.co.uk>

### References

Hawkins, D. , Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31

Hawkins, D. , Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173

Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.

Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)

Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116

Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)

Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030

Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

### See Also

[processObservation](#), [changeDetected](#), [cpmReset](#).

### Examples

```
#generate a sequence containing a single change point
x <- c(rnorm(100,0,1),rnorm(100,1,1))

#use a Student CPM
cpm <- makeChangePointModel(cpmType="Student", ARL0=500)

for (i in 1:length(x)) {
```

```
#process each observation in turn
cpm <- processObservation(cpm,x[i])

if (changeDetected(cpm)) {
  print(sprintf("change detected at observation %s",i))
  break
}
}
```

---

processObservation      *Process an Observation Using a CPM S4 Object.*

---

### Description

Updates the state of an existing Change Point Model (CPM) S4 object, by processing a single observation. This effectively computes the  $D_{k,t+1}$  statistics, for a CPM that had previously seen  $t$  observations. When the function is called, several events happen. First, the function returns a CPM object which is identical the CPM object passed to the function, except that the observation passed as an argument has been processed and added to the state. Second, the CPM computes the  $D_{t+1}$  statistic and compares it to its stored sequence of thresholds. If a change is detected, then this is stored in the state of the CPM, and a call to `changeDetected` will now return TRUE.

Note that this function is part of the S4 object section of the `cpm` package, which allows for more precise control over the change detection process. For many simple change detection applications this extra complexity will not be required, and the `detectChangePoint` and `processStream` functions should be used instead.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The `cpm` Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

### Usage

```
processObservation(cpm,x)
```

### Arguments

<code>cpm</code>	The CPM S4 object which is to be updated.
<code>x</code>	The observation which is to be processed.

### Value

The updated CPM. If a stream is being processed, then this should be stored, and used to process the next observation in the sequence.

**Author(s)**

Gordon J. Ross <gordon@gordonjross.co.uk>

**See Also**

[makeChangePointModel](#), [changeDetected](#).

**Examples**

```
#generate a sequence containing a single change point
x <- c(rnorm(100,0,1),rnorm(100,1,1))

#use a Student CPM
cpm <- makeChangePointModel(cpmType="Student", ARL0=500)

for (i in 1:length(x)) {

  #process each observation in turn
  cpm <- processObservation(cpm,x[i])

  if (changeDetected(cpm)) {
    print(sprintf("change detected at observation %s",i))
    break
  }
}
```

---

processStream

*Detects Multiple Change Points in a Sequence*

---

**Description**

This function is used to detect a multiple change points in a sequence of observations using the Change Point Model (CPM) framework for sequential (Phase II) change detection. The observations are processed in order, starting with the first, and a decision is made after each observation whether a change point has occurred. A full description of the CPM framework can be found in the papers cited in the reference section.

Unlike the `detectChange` function, `processStream` does not terminate and return when a change point is encountered. Instead, a new CPM is initialised immediately following the change point, with all previous observations being discarded. The monitoring then continues, starting from the first observation after the change point. If more change points are discovered later in the sequence, the CPM is again reinitialised after each one. In this way, the whole sequence of observations will be processed and multiple change points may be detected.

For a fuller overview of this function including a description of the CPM framework and examples of how to use the various functions, please consult the package manual "Parametric and Nonparametric Sequential Change Detection in R: The cpm Package" available from [www.gordonjross.co.uk](http://www.gordonjross.co.uk)

**Usage**

```
processStream(x, cpmType, ARL0=500, startup=20, lambda=NA)
```

**Arguments**

x	A vector containing the univariate data stream to be processed.
cpmType	<p>The type of CPM which is used. Possible arguments are:</p> <ul style="list-style-type: none"> <li>• <b>Student</b>: Student-t test statistic, as in [Hawkins et al, 2003]. Use to detect mean changes in a Gaussian sequence.</li> <li>• <b>Bartlett</b>: Bartlett test statistic, as in [Hawkins and Zamba, 2005]. Use to detect variance changes in a Gaussian sequence.</li> <li>• <b>GLR</b>: Generalized Likelihood Ratio test statistic, as in [Hawkins and Zamba, 2005b]. Use to detect both mean and variance changes in a Gaussian sequence.</li> <li>• <b>Exponential</b>: Generalized Likelihood Ratio test statistic for the Exponential distribution, as in [Ross, 2013]. Used to detect changes in the parameter of an Exponentially distributed sequence.</li> <li>• <b>GLRAdjusted</b> and <b>ExponentialAdjusted</b>: Identical to the GLR and Exponential statistics, except with the finite-sample correction discussed in [Ross, 2013] which can lead to more powerful change detection.</li> <li>• <b>FET</b>: Fishers Exact Test statistic, as in [Ross and Adams, 2012b]. Use to detect parameter changes in a Bernoulli sequence.</li> <li>• <b>Mann-Whitney</b>: Mann-Whitney test statistic, as in [Ross et al, 2011]. Use to detect location shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Mood</b>: Mood test statistic, as in [Ross et al, 2011]. Use to detect scale shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Lepage</b>: Lepage test statistics in [Ross et al, 2011]. Use to detect location and/or shifts in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Kolmogorov-Smirnov</b>: Kolmogorov-Smirnov test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> <li>• <b>Cramer-von-Mises</b>: Cramer-von-Mises test statistic, as in [Ross et al 2012]. Use to detect arbitrary changes in a stream with a (possibly unknown) non-Gaussian distribution.</li> </ul>
ARL0	Determines the $ARL_0$ which the CPM should have, which corresponds to the average number of observations before a false positive occurs, assuming that the sequence does not undergo a change. Because the thresholds of the CPM are computationally expensive to estimate, the package contains pre-computed values of the thresholds corresponding to several common values of the $ARL_0$ . This means that only certain values for the $ARL_0$ are allowed. Specifically, the $ARL_0$ must have one of the following values: 370, 500, 600, 700, ..., 1000, 2000, 3000, ..., 10000, 20000, ..., 50000.

startup	The number of observations after which monitoring begins. No change points will be flagged during this startup period. This should be set to at least 20.
lambda	A smoothing parameter which is used to reduce the discreteness of the test statistic when using the FET CPM. See [Ross and Adams, 2012b] in the References section for more details on how this parameter is used. Currently the package only contains sequences of ARL0 thresholds corresponding to lambda=0.1 and lambda=0.3, so using other values will result in an error. If no value is specified, the default value will be 0.1.

**Value**

x	The sequence of observations which was processed.
detectionTimes	A vector containing the points in the sequence at which changes were detected, defined as the first observation after which $D_t$ exceeded the test threshold.
changePoints	A vector containing the best estimates of the change point locations, for each detecting change point. If a change is detected after the $t^{th}$ observation, then the change estimate is the value of $k$ which maximises $D_{k,t}$ .

**Author(s)**

Gordon J. Ross <gordon@gordonjross.co.uk>

**References**

- Hawkins, D. , Zamba, K. (2005) – A Change-Point Model for a Shift in Variance, *Journal of Quality Technology*, 37, 21-31
- Hawkins, D. , Zamba, K. (2005b) – Statistical Process Control for Shifts in Mean or Variance Using a Changepoint Formulation, *Technometrics*, 47(2), 164-173
- Hawkins, D., Qiu, P., Kang, C. (2003) – The Changepoint Model for Statistical Process Control, *Journal of Quality Technology*, 35, 355-366.
- Ross, G. J., Tasoulis, D. K., Adams, N. M. (2011) – A Nonparametric Change-Point Model for Streaming Data, *Technometrics*, 53(4)
- Ross, G. J., Adams, N. M. (2012) – Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes, *Journal of Quality Technology*, 44:102-116
- Ross, G. J., Adams, N. M. (2013) – Sequential Monitoring of a Proportion, *Computational Statistics*, 28(2)
- Ross, G. J., (2014) – Sequential Change Detection in the Presence of Unknown Parameters, *Statistics and Computing* 24:1017-1030
- Ross, G. J., (2015) – Parametric and Nonparametric Sequential Change Detection in R: The cpm Package, *Journal of Statistical Software*, forthcoming

**See Also**

[detectChangePoint.](#)

**Examples**

```
## Use a Student-t CPM to detect several mean shift in a stream of
## Gaussian random variables
x <- c(rnorm(100,0,1),rnorm(100,1,1), rnorm(100,0,1), rnorm(100,-1,1))
result <- processStream(x,"Student",ARL0=500,startup=20)
plot(x)
for (i in 1:length(result$changePoints)) {
  abline(v=result$changePoints[i], lty=2)
}

## Use a Mood CPM to detect several scale shifts in a stream of
## Student-t random variables
x <- c(rt(100,3),rt(100,3)*2, rt(100,3), rt(100,3)*2)
result <- processStream(x,"Mood",ARL0=500,startup=20)
plot(x)
for (i in 1:length(result$changePoints)) {
  abline(v=result$changePoints[i], lty=2)
}

## Use a FET CPM to detect several parameter shifts in a stream of
## Bernoulli observations. In this case, the lambda parameter acts to
## reduce the discreteness of the test statistic.
x <- c(rbinom(300,1,0.1),rbinom(300,1,0.4), rbinom(300,1,0.7))
result <- processStream(x,"FET",ARL0=500,startup=20,lambda=0.3)
plot(x)
for (i in 1:length(result$changePoints)) {
  abline(v=result$changePoints[i], lty=2)
}
```

# Index

## \*Topic **datasets**

ForexData, [12](#)

changeDetected, [4](#), [5](#), [15](#), [18](#), [20](#)

cpm-package, [2](#)

cpmReset, [5](#), [18](#)

detectChangePoint, [6](#), [11](#), [22](#)

detectChangePointBatch, [9](#), [9](#), [14](#)

ForexData, [12](#)

getBatchThreshold, [13](#)

getStatistics, [15](#)

makeChangePointModel, [4](#), [5](#), [15](#), [16](#), [20](#)

processObservation, [4](#), [5](#), [15](#), [18](#), [19](#)

processStream, [9](#), [20](#)