



## e`cp`: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data

Nicholas A. James  
Cornell University

David S. Matteson  
Cornell University

---

### Abstract

There are many different ways in which change point analysis can be performed, from purely parametric methods to those that are distribution free. The `ecp` package is designed to perform multiple change point analysis while making as few assumptions as possible. While many other change point methods are applicable only for univariate data, this R package is suitable for both univariate and multivariate observations. Hierarchical estimation can be based upon either a divisive or agglomerative algorithm. Divisive estimation sequentially identifies change points via a bisection algorithm. The agglomerative algorithm estimates change point locations by determining an optimal segmentation. Both approaches are able to detect *any* type of distributional change within the data. This provides an advantage over many existing change point algorithms which are only able to detect changes within the marginal distributions.

*Keywords:* cluster analysis, multivariate time series, signal processing.

---

## 1. Introduction

Change point analysis is the process of detecting distributional changes within time-ordered observations. This arises in financial modeling (Talih and Hengartner 2005), where correlated assets are traded and models are based on historical data. It is applied in bioinformatics (Muggeo and Adelfio 2011) to identify genes that are associated with specific cancers and other diseases. Change point analysis is also used to detect credit card fraud (Bolton and Hand 2002) and other anomalies (Akoglu and Faloutsos 2010; Sequeira and Zaki 2002); and for data classification in data mining (Mampaey and Vreeken 2013).

We introduce the R (R Core Team 2014) package `ecp` for multiple change point analysis of multivariate time series (Matteson and James 2014). The `ecp` package provides methods for change point analysis that are able to detect *any* type of distributional change within

a time series. Determination of the number of change points is also addressed by these methods as they estimate both the number and locations of change points simultaneously. The only assumptions placed on distributions are that the absolute  $\alpha$ th moment exists, for some  $\alpha \in (0, 2]$ , and that observations are independent over time. Distributional changes are identified by making use of the energy statistic of Székely and Rizzo (2005, 2010).

There are a number of freely available R packages that can be used to perform change point analysis, each making its own assumptions about the observed time series. For instance, the **changepoint** package (Killick and Eckley 2014) provides many methods for performing change point analysis of univariate time series. Although the package only considers the case of independent observations, the theory behind the implemented methods allows for certain types of serial dependence (Killick, Fearnhead, and Eckley 2012). For specific methods, the expected computational cost can be shown to be linear with respect to the length of the time series. Currently, the **changepoint** package is only suitable for finding changes in mean or variance. This package also estimates multiple change points through the use of penalization. The drawback to this approach is that it requires a user specified penalty term.

The **cpm** package (Ross 2013) similarly provides a variety of methods for performing change point analysis of univariate time series. These methods range from those to detect changes in independent Gaussian data to fully nonparametric methods that can detect general distributional changes. Although this package provides methods to perform analysis of univariate time series with arbitrary distributions, these methods cannot be easily extended to detect changes in the full joint distribution of multivariate data.

Unlike the **changepoint** and **cpm** packages, the **bcp** package (Erdman and Emerson 2007) is designed to perform Bayesian change point analysis of univariate time series. It returns the posterior probability of a change point occurring at each time index in the series. Recent versions of this package have reduced the computational cost from quadratic to linear with respect to the length of the series. However, all versions of this package are only designed to detect changes in the mean of independent Gaussian observations.

The **strucchange** package (Zeileis, Leisch, Hornik, and Kleiber 2002) provides a suite of tools for detecting changes within linear regression models, methods that perform online change detection. Additionally, the **breakpoints** method (Zeileis, Kleiber, Krämer, and Hornik 2003) can be used to perform retrospective analysis. For a given number of changes, this method returns the change point estimates which minimize the residual sum of squares and the number of change points can subsequently be chosen using information criteria.

In Section 2 we introduce the energy statistic of Székely and Rizzo (2005, 2010), which is the fundamental divergence measure applied for change point analysis. Sections 3 and 4 briefly outline the package's methods. Section 5 provide examples of these methods being applied to simulated data, while Section 6 presents applications to real datasets. In the Appendix we include an outline of the algorithms used by this package's methods. Finally, the **ecp** package can be freely obtained from the Comprehensive R Archive Network (CRAN) at <http://CRAN.R-project.org/package=ecp>.

## 2. The **ecp** package

The **ecp** package is designed to address many of the limitations of the currently available change point packages. It is able to perform multiple change point analysis for both uni-

variate and multivariate time series. The methods are able to estimate multiple change point locations, without *a priori* knowledge of the number of change points. The procedures assume that observations are independent with finite  $\alpha$ th absolute moments, for some  $\alpha \in (0, 2]$ .

## 2.1. Measuring differences in multivariate distributions

Székely and Rizzo (2005, 2010) introduce a divergence measure that can determine whether two independent random vectors are identically distributed. Suppose that  $X, Y \in \mathbb{R}^d$  are such that,  $X \sim F$  and  $Y \sim G$ , with characteristic functions  $\phi_x(t)$  and  $\phi_y(t)$ , respectively. A divergence measure between the two distributions may be defined as

$$\int_{\mathbb{R}^d} |\phi_x(t) - \phi_y(t)|^2 w(t) dt,$$

in which  $w(t)$  is any positive weight function, for which the above integral is defined. Following Matteson and James (2014) we employ the following weight function,

$$w(t; \alpha) = \left( \frac{2\pi^{d/2}\Gamma(1 - \alpha/2)}{\alpha 2^\alpha \Gamma[(d + \alpha)/2]} |t|^{d+\alpha} \right)^{-1},$$

for some fixed constant  $\alpha \in (0, 2)$ . Thus our divergence measure is

$$\mathcal{D}(X, Y; \alpha) = \int_{\mathbb{R}^d} |\phi_x(t) - \phi_y(t)|^2 \left( \frac{2\pi^{d/2}\Gamma(1 - \alpha/2)}{\alpha 2^\alpha \Gamma[(d + \alpha)/2]} |t|^{d+\alpha} \right)^{-1} dt.$$

An alternative divergence measure based on Euclidean distances may be defined as follows

$$\mathcal{E}(X, Y; \alpha) = 2\mathbb{E}|X - Y|^\alpha - \mathbb{E}|X - X'|^\alpha - \mathbb{E}|Y - Y'|^\alpha.$$

In the above equation,  $X'$  and  $Y'$  are independent copies of  $X$  and  $Y$ , respectively. Then given our choice of weight function, we have the following result.

**Lemma 1.** *For any pair of independent random variables  $X, Y \in \mathbb{R}^d$  and for any  $\alpha \in (0, 2)$ , if  $\mathbb{E}(|X|^\alpha + |Y|^\alpha) < \infty$ , then  $\mathcal{D}(X, Y; \alpha) = \mathcal{E}(X, Y; \alpha)$ ,  $\mathcal{E}(X, Y; \alpha) \in [0, \infty)$ , and  $\mathcal{E}(X, Y; \alpha) = 0$  if and only if  $X$  and  $Y$  are identically distributed.*

*Proof.* A proof is given in the appendices of Székely and Rizzo (2005) and Matteson and James (2014). □

Thus far we have assumed that  $\alpha \in (0, 2)$ , because in this setting  $\mathcal{E}(X, Y; \alpha) = 0$  if and only if  $X$  and  $Y$  are identically distributed. However, if we allow for  $\alpha = 2$  a weaker result of equality in mean is obtained.

**Lemma 2.** *For any pair of independent random variables  $X, Y \in \mathbb{R}^d$ , if  $\mathbb{E}(|X|^2 + |Y|^2) < \infty$ , then  $\mathcal{D}(X, Y; 2) = \mathcal{E}(X, Y; 2)$ ,  $\mathcal{E}(X, Y; 2) \in [0, \infty)$ , and  $\mathcal{E}(X, Y; 2) = 0$  if and only if  $\mathbb{E}X = \mathbb{E}Y$ .*

*Proof.* See Székely and Rizzo (2005). □

## 2.2. A sample divergence for multivariate distributions

Let  $X \sim F$  and  $Y \sim G$  for arbitrary distributions  $F$  and  $G$ . Additionally, select  $\alpha \in (0, 2)$  such that  $E|X|^\alpha, E|Y|^\alpha < \infty$ . Let  $\mathbf{X}_n = \{X_i : i = 1, 2, \dots, n\}$  be  $n$  independent observations with  $X_i \sim F$ , and  $\mathbf{Y}_m = \{Y_j : j = 1, \dots, m\}$  are  $m$  independent observations with  $Y_j \sim G$ . Furthermore, we assume full mutual independence between all observations,  $\mathbf{X}_n \perp\!\!\!\perp \mathbf{Y}_m$ . Then Lemmas 1 and 2 suggest following sample divergence measure,

$$\widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha) = \frac{2}{mn} \sum_{i=1}^n \sum_{j=1}^m |X_i - Y_j|^\alpha - \binom{n}{2}^{-1} \sum_{1 \leq i < k \leq n} |X_i - X_k|^\alpha - \binom{m}{2}^{-1} \sum_{1 \leq j < k \leq m} |Y_j - Y_k|^\alpha. \quad (1)$$

By the strong law of large numbers for  $U$ -statistics (Hoeffding 1961)  $\widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha) \xrightarrow{a.s.} \mathcal{E}(X, Y; \alpha)$  as  $n \wedge m \rightarrow \infty$ . Equation 1 allows for an estimate of  $\mathcal{D}(X, Y; \alpha)$  without performing  $d$ -dimensional integration. Furthermore, let

$$\widehat{\mathcal{Q}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha) = \frac{mn}{m+n} \widehat{\mathcal{E}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha) \quad (2)$$

denote the scaled empirical divergence. Under the null hypothesis of equal distributions, i.e.,  $\mathcal{E}(X, Y; \alpha) = 0$ , Székely and Rizzo (2010) show that  $\widehat{\mathcal{Q}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha)$  converges in distribution to a non-degenerate random variable  $\mathcal{Q}(X, Y; \alpha)$  as  $m \wedge n \rightarrow \infty$ . Specifically,

$$\mathcal{Q}(X, Y; \alpha) = \sum_{i=1}^{\infty} \lambda_i Q_i$$

in which the  $\lambda_i \geq 0$  are constants that depend on  $\alpha$  and the distributions of  $X$  and  $Y$ , and the  $Q_i$  are iid chi-squared random variables with one degree of freedom. Under the alternative hypothesis of unequal distributions, i.e.,  $\mathcal{E}(X, Y; \alpha) > 0$ ,  $\widehat{\mathcal{Q}}(\mathbf{X}_n, \mathbf{Y}_m; \alpha) \rightarrow \infty$  almost surely as  $m \wedge n \rightarrow \infty$ .

Gretton, Borgwardt, Rasch, Schölkopf, and Smola (2007) use a statistic similar to that presented in Equation 1 to test for equality in distribution. This alternative statistic is used by Schauer, Duong, Bleakley, Bardin, Bornens, and Goud (2010) to test whether the point set inside a new perturbed set of cells had a different distribution to the ‘‘unperturbed’’ cells. However, this statistic, unlike the one presented in Equation 1, is only applicable when the unknown distributions have continuous bounded density functions.

By using Equations 1 and 2, along with their convergence properties, we are able to develop two hierarchical methods for performing change point analysis. Section 3 presents our divisive method, while our agglomerative method is presented in Section 4.

## 3. Hierarchical divisive estimation: E-Divisive

We first present the E-Divisive method for performing hierarchical divisive estimation of multiple change points. Here, multiple change points are estimated by iteratively applying a procedure for locating a single change point. At each iteration a new change point location is estimated so that it divides an existing segment. As a result, the progression of this method can be diagrammed as a binary tree. In this tree, the root node corresponds to the case of no change points, and thus contains the entire time series. All other non-root nodes are either a copy of their parent, or correspond to one of the new segments created by the addition of

a change point to their parent. Details on the estimation of change point locations can be found in [Matteson and James \(2014\)](#).

The statistical significance of an estimated change point is determined through a permutation test, since the distribution of the test statistic depends upon the distributions of the observations, which is *unknown* in general. Suppose that at the  $k$ th iteration the current set of change points has segmented the time series in the  $k$  segments  $S_1, S_2, \dots, S_k$ , and that we have estimated the next change point location as  $\hat{\tau}_k$ , which has an associated test statistic value of  $q_0$ . We then obtain our permuted sample by permuting the observations within each of  $S_1, \dots, S_k$ . Then, conditional on the previously estimated change point locations, we estimate the location of the next change point in our permuted sample,  $\hat{\tau}_{k,r}$ , along with its associated testing statistic value  $q_r$ . Our approximate  $p$  value is then calculated as  $\hat{p} = \#\{r : q_r \geq q_0\} / (R + 1)$ , where  $R$  is the total number of permutations performed.

The signature of the method used to perform analysis based on this divisive approach is

```
e.divisive(X, sig.lvl = 0.05, R = 199, k = NULL, min.size = 30, alpha = 1)
```

Descriptions for all function arguments can be found in the package's help files. The time complexity of this method is  $\mathcal{O}(kT^2)$ , where  $k$  is the number of estimated change points, and  $T$  is the number of observations in the series. Due to the running time being quadratic in the length of the time series this procedure is not recommended for series larger than several thousand observations. Future versions of the package will provide tools that can be used to reduce the computational time. One such tool will be to obtain a permutation  $p$  value with a uniformly bounded resampling risk ([Gandy 2009](#)).

In the case of independent observations, [Matteson and James \(2014\)](#) show that this procedure generates strongly consistent change point estimates. There are other faster approaches for performing nonparametric multiple change point analysis, however they do not have a similar consistency guarantee, which is why we recommend our divisive approach when appropriate. A more complete outline of the divisive algorithm is detailed in the [Appendix A.1](#).

## 4. Hierarchical agglomerative estimation: E-Aggl

We now present the E-Aggl method for performing hierarchical agglomerative estimation of multiple change points. This method requires that an initial segmentation of the data be provided. This initial segmentation can help to reduce the computational time of the procedure. It also allows for the inclusion of *a priori* knowledge of possible change point locations, however if no such assumptions are made, then each observation can be assigned to its own segment. Neighboring segments are then sequentially merged to maximize a goodness-of-fit statistic. The estimated change point locations are determined by the iteration which maximized the penalized goodness-of-fit statistic. When using the E-Aggl procedure it is assumed that there is at least one change point present within the time series.

The goodness-of-fit statistic used in [Matteson and James \(2014\)](#) is the between-within distance ([Székely and Rizzo 2005](#)) among adjacent segments. Let  $\mathcal{C} = \{C_1, \dots, C_n\}$  be a segmentation of the  $T$  observations into  $n$  segments. The goodness-of-fit statistic is defined as

$$\hat{S}_n(\mathcal{C}; \alpha) = \sum_{i=1}^{n-1} \hat{Q}(C_i, C_{i+1}; \alpha). \quad (3)$$

Since calculating the true maximum of the goodness-of-fit statistic for a given initial segmentation would be too computationally intensive, a greedy algorithm is used to find an approximate solution. For a detailed explanation of how this algorithm is efficiently carried out see the Appendix A.2.

If overfitting is a concern, it is possible to penalize the sequence of goodness-of-fit statistics. This is accomplished through the use of the `penalty` argument, which generates a penalty based upon change point locations. Thus, the change point locations are estimated by maximizing

$$\tilde{\mathcal{S}}_k = \hat{\mathcal{S}}_k + \text{penalty}(\vec{\tau}(k)),$$

where  $\vec{\tau}(k) = \{\tau_1, \tau_2, \dots, \tau_k\}$  is the set of change points associated with the goodness-of-fit statistic  $\hat{\mathcal{S}}_k$ . Examples of penalty terms include

```
penalty1 <- function(cp) -length(cp)
penalty2 <- function(cp) mean(diff(sort(cp)))
```

Here `penalty1` corresponds to the function  $\text{penalty}(\vec{\tau}(k)) = -k$  while `penalty2` corresponds to the function

$$\text{penalty}(\vec{\tau}(k)) = \frac{1}{k+1} \sum_{i=1}^{k-1} [\tau_{i+1} - \tau_i].$$

Both penalties favor segmentations with larger sizes. However, `penalty1` equally penalizes every additional change point, while `penalty2` takes the size of the new segments into consideration.

The signature of the method used to perform agglomerative analysis is

```
e.agglo(X, member = 1:nrow(X), alpha = 1, penalty = function(cp) 0)
```

Descriptions for all function arguments can be found in the package's help files. Like the E-Divisive method, this is quadratic in the number of observations with computational complexity  $\mathcal{O}(T^2)$ , however its complexity does not depend on the number of estimated change points.

## 5. Examples

In this section we illustrate the use of both the `e.divisive` and `e.agglo` functions to perform multivariate change point analysis. Both functions will return a list of estimated change points,  $\{\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_k\}$ , as well as additional information. For both functions, if  $\hat{\tau}_1 = 1$ , then the method has determined that the observations within the interval  $[\hat{\tau}_i, \hat{\tau}_{i+1})$  are identically distributed, for  $i = 1, 2, \dots, k-1$ . However, in the case of the `e.agglo` function, if  $\hat{\tau}_1 \neq 1$ , the method has determined that the observations in  $[\hat{\tau}_i, \hat{\tau}_{i+1})$  are identically distributed, for  $i = 1, \dots, k-1$ , while  $[1, \hat{\tau}_1)$  and  $[\hat{\tau}_{k-1}, \hat{\tau}_k)$  are identically distributed.

### 5.1. Change in univariate normal distribution

We begin with the simple case of identifying change in univariate normal distributions. For this we sequentially generate 100 independent samples from the following normal distributions:  $\mathcal{N}(0, 1)$ ,  $\mathcal{N}(0, \sqrt{3})$ ,  $\mathcal{N}(2, 1)$ , and  $\mathcal{N}(2, 2)$ .

```

R> set.seed(250)
R> library("ecp")
R> period1 <- rnorm(100)
R> period2 <- rnorm(100, 0, 3)
R> period3 <- rnorm(100, 2, 1)
R> period4 <- rnorm(100, 2, 4)
R> Xnorm <- matrix(c(period1, period2, period3, period4), ncol = 1)
R> output1 <- e.divisive(Xnorm, R = 499, alpha = 1)
R> output2 <- e.divisive(Xnorm, R = 499, alpha = 2)
R> output2$estimates

[1] 1 201 358 401

R> output1$k.hat

[1] 4

R> output1$order.found

[1] 1 401 201 308 108

R> output1$estimates

[1] 1 108 201 308 401

R> output1$considered.last

[1] 358

R> output1$p.values

[1] 0.002 0.002 0.010 0.916

R> output1$permutations

[1] 499 499 499 499

R> ts.plot(Xnorm, ylab = "Value",
+ main = "Change in a Univariate Gaussian Sequence")
R> abline(v = c(101, 201, 301), col = "blue")
R> abline(v = output1$estimates[c(-1, -5)], col = "red", lty = 2)

```

As can be seen, if  $\alpha = 2$  the E-Divisive method can only identify changes in mean. For this reason, it is recommended that  $\alpha$  is selected so as to lie in the interval  $(0, 2)$ , in general. Figure 1 depicts the example time series, along with the estimated change points from the E-Divisive method when using  $\alpha = 1$ .

Furthermore, when applying the E-Aggllo method to this same simulated dataset, we obtain similar results. In the R code below we present the case for  $\alpha = 1$ .

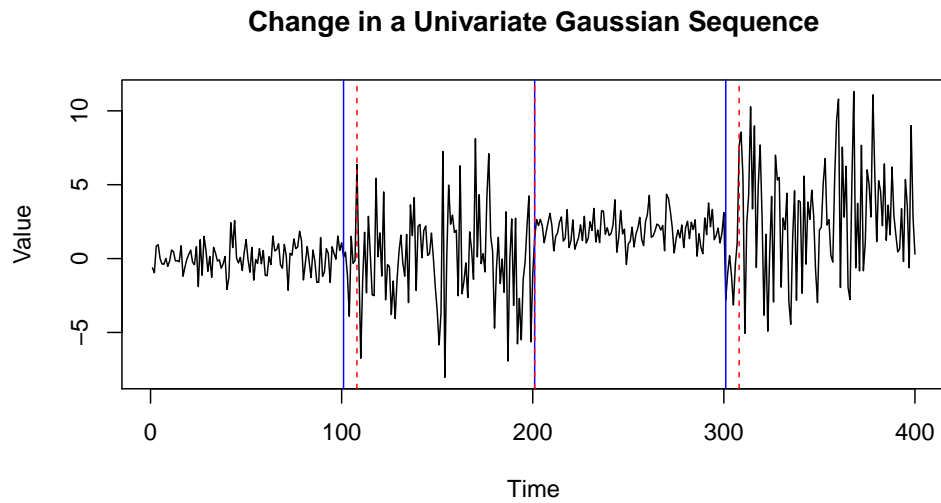


Figure 1: Simulated independent Gaussian observations with changes in mean or variance. Dashed vertical lines indicate the change point locations estimated by the E-Divisive method, when using  $\alpha = 1$ . Solid vertical lines indicate the true change point locations.

```
R> member <- rep(1:40, each = 10)
R> output <- e.agglo(X = Xnorm, member = member, alpha = 1)
R> output$estimates

[1] 1 101 201 301 401

R> tail(output$fit, 5)

[1] 100.05695 107.82542 104.30608 102.64330 -17.10722

R> output$progression[1, 1:10]

[1] 1 11 21 31 41 51 61 71 81 91

R> output$merged[1:4, ]

      [,1] [,2]
[1,]  -39 -40
[2,]   -1  -2
[3,]  -38  1
[4,]   2  -3
```

## 5.2. Multivariate change in covariance

To demonstrate that our methods do not just identify changes in marginal distributions we consider a multivariate example with only a change in covariance. In this example the



marginal distributions remain the same, but the joint distribution changes. Therefore, applying a univariate change point procedure to each margin, such as those implemented by the **changepoint**, **cpm**, and **bcp** packages, will not detect the changes. The observations in this example are drawn from trivariate normal distributions with mean vector  $\mu = (0, 0, 0)^\top$  and the following covariance matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Observations are generated by using the **mvtnorm** package (Genz, Bretz, Miwa, Mi, Leisch, Scheipl, and Hothorn 2014).

```
R> set.seed(200)
R> library("mvtnorm")
R> mu <- rep(0, 3)
R> covA <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 1), 3, 3)
R> covB <- matrix(c(1, 0.9, 0.9, 0.9, 1, 0.9, 0.9, 0.9, 1), 3, 3)
R> period1 <- rmvnorm(250, mu, covA)
R> period2 <- rmvnorm(250, mu, covB)
R> period3 <- rmvnorm(250, mu, covA)
R> Xcov <- rbind(period1, period2, period3)
R> DivOutput <- e.divisive(Xcov, R = 499, alpha = 1)
R> DivOutput$estimates

[1] 1 250 502 751

R> member <- rep(1:15, each = 50)
R> pen <- function(x) -length(x)
R> AggOutput1 <- e.agglo(X = Xcov, member = member, alpha = 1)
R> AggOutput2 <- e.agglo(X = Xcov, member = member, alpha = 1, penalty = pen)
R> AggOutput1$estimates

[1] 1 101 201 301 351 501 601 701 751

R> AggOutput2$estimates

[1] 301 501
```

In this case, the default procedure generates too many change points, as can be seen by the result of `AggOutput1`. When penalizing based upon the number of change points we obtain a much more accurate result, as shown by `AggOutput2`. Here the E-Agglo method has indicated that observations 1 through 300 and observations 501 through 750 are identically distributed.

### 5.3. Multivariate change in tails

For our second multivariate example we consider the case where the change in distribution is caused by a change in tail behavior. Data points are drawn from a bivariate normal distribution and a bivariate  $t$  distribution with 2 degrees of freedom. Figure 2 depicts the different samples within the time series.

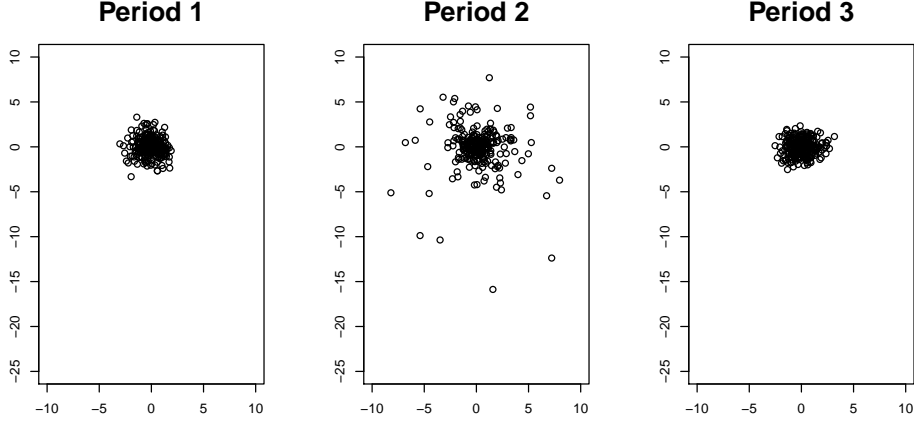


Figure 2: Data set used for the change in tail behavior example from Section 5.3. Periods 1 and 3 contain independent bivariate Gaussian observations with mean vector  $(0, 0)^\top$  and identity covariance matrix. The second time period contains independent observations from a bivariate Student's  $t$  distribution with 2 degrees of freedom and identity covariance matrix.

```
R> set.seed(100)
R> mu <- rep(0, 2)
R> period1 <- rmvnorm(250, mu, diag(2))
R> period2 <- rmvt(250, sigma = diag(2), df = 2)
R> period3 <- rmvnorm(250, mu, diag(2))
R> Xtail <- rbind(period1, period2, period3)
R> output <- e.divisive(Xtail, R = 499, alpha = 1)
R> output$estimates
```

```
[1] 1 257 504 751
```

#### 5.4. Inhomogeneous spatio-temporal point process

We apply the E-Aggl procedure to a spatio-temporal point process. The examined dataset consists of 10,498 observations, each with associated time and spatial coordinates. This dataset spans the time interval  $[0, 7]$  and has spatial domain  $\mathbb{R}^2$ . It contains 3 change points, which occur at times  $t_1 = 1$ ,  $t_2 = 3$ , and  $t_3 = 4.5$ . Over each of these subintervals,  $t \in [t_i, t_{i+1}]$  the process is an inhomogeneous Poisson point process with intensity function  $\lambda(s, t) = f_i(s)$ , a 2-d density function, for  $i = 1, 2, 3, 4$ . This intensity function is chosen to be the density function from a mixture of 3 bivariate normal distributions,

$$\mathcal{N}\left(\begin{pmatrix} -7 \\ -7 \end{pmatrix}, \begin{pmatrix} 25 & 0 \\ 0 & 25 \end{pmatrix}\right), \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}\right), \text{ and } \mathcal{N}\left(\begin{pmatrix} 5.5 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 & 0.9 \\ 0.9 & 9 \end{pmatrix}\right).$$

For the time periods,  $[0, 1]$ ,  $(1, 3]$ ,  $(3, 4.5]$ , and  $(4.5, 7]$  the respective mixture weights are

$$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \left(\frac{1}{5}, \frac{1}{2}, \frac{3}{10}\right), \left(\frac{7}{20}, \frac{3}{10}, \frac{7}{20}\right), \text{ and } \left(\frac{1}{5}, \frac{3}{10}, \frac{1}{2}\right).$$

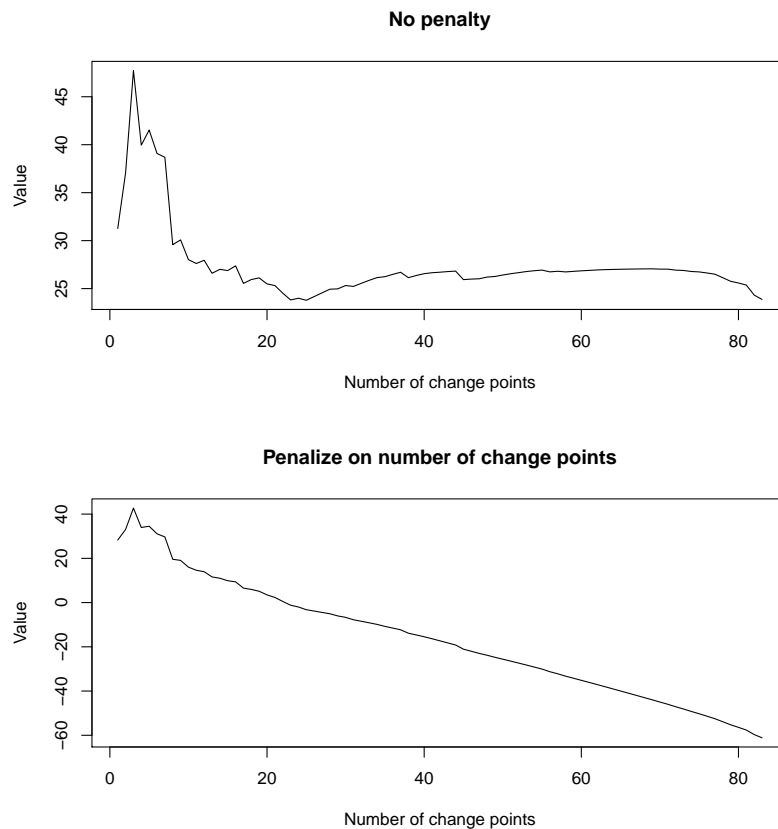


Figure 3: The progression of the goodness-of-fit statistic for the various penalization schemes discussed in Section 5.4.

To apply the E-Aggl procedure we initially segment the observations into 50 segments such that each segment spans an equal amount of time. At its termination, the E-Aggl procedure, with no penalty, identified change points at times 0.998, 3.000, and 4.499. These results can be obtained with the following code, where `lambda` is the overall arrival rate per unit time and packages `combinat` (Chasalow 2012) and `MASS` (Venables and Ripley 2002) need to be loaded.

```
R> library("combinat")
R> library("MASS")
R> set.seed(2013)
R> lambda <- 1500
R> muA <- c(-7, -7)
R> muB <- c(0, 0)
R> muC <- c(5.5, 0)
R> covA <- 25 * diag(2)
R> covB <- matrix(c(9, 0, 0, 1), 2)
R> covC <- matrix(c(9, 0.9, 0.9, 9), 2)
R> time.interval <- matrix(c(0, 1, 3, 4.5, 1, 3, 4.5, 7), 4, 2)
R> mixing.coef <- rbind(c(1/3, 1/3, 1/3), c(0.2, 0.5, 0.3),
+   c(0.35, 0.3, 0.35), c(0.2, 0.3, 0.5))
```

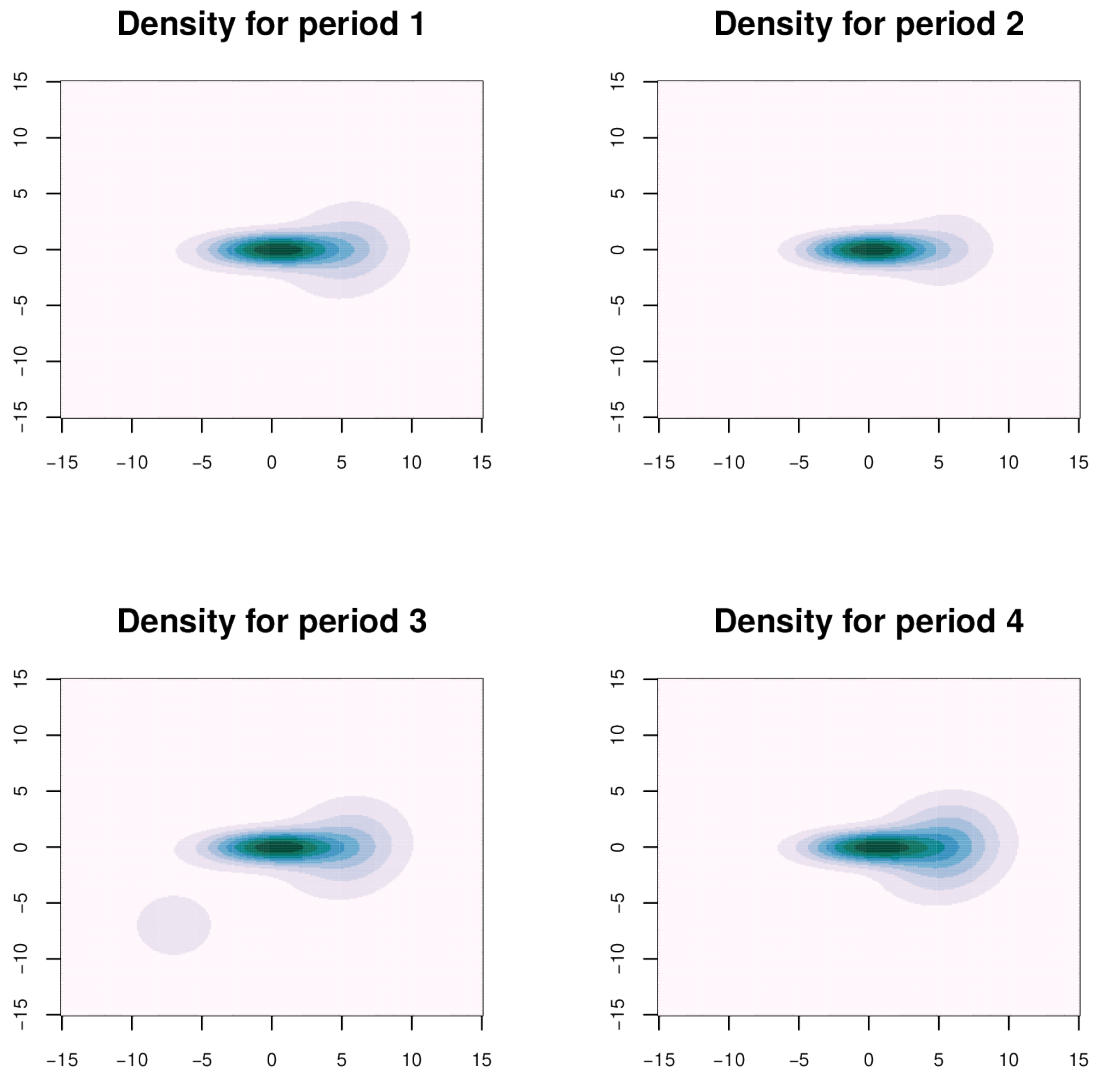


Figure 4: True density plots for the different segments of the spatio-temporal point process in Section 5.4.

```
R> stppData <- NULL
R> for (i in 1:4) {
+   count <- rpois(1, lambda * diff(time.interval[i, ]))
+   Z <- rmultz2(n = count, p = mixing.coef[i, ])
+   S <- rbind(rmvnorm(Z[1], muA, covA), rmvnorm(Z[2], muB, covB),
+             rmvnorm(Z[3], muC, covC))
+   X <- cbind(rep(i, count), runif(n = count, time.interval[i, 1],
+   time.interval[i, 2]), S)
+   stppData <- rbind(stppData, X[order(X[, 2]), ])
+ }
R> member <- as.numeric(cut(stppData[, 2], breaks = seq(0, 7, by = 1/12)))
R> output <- e.agglo(X = stppData[, 3:4], member = member, alpha = 1)
```

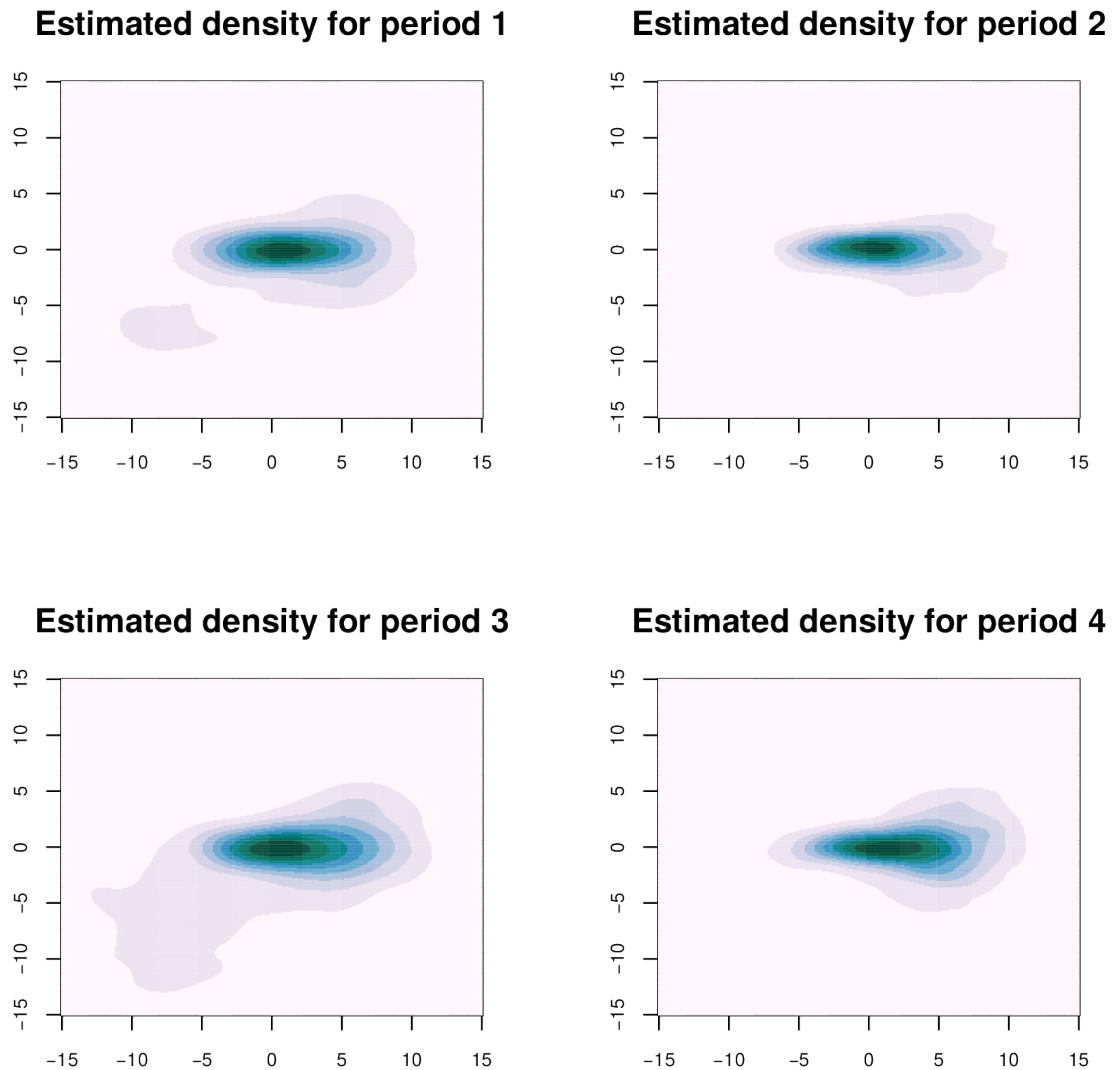


Figure 5: Estimated density plots for the estimated segmentation provided by the E-Agglo procedure when applied to the spatio-temporal point process in Section 5.4.

The E-Agglo procedure was also run on the above data set using the following penalty function,

```
pen <- function(cp) -length(cp)
```

When using `pen`, change points were also estimated at times 0.998, 3.000, 4.499. The progression of the goodness-of-fit statistic for the different schemes is plotted in Figure 3. A comparison of the true densities and the estimated densities obtained from the procedure's results with no penalty are shown in Figures 4 and 5, respectively. As can be seen, the estimated results obtained from the E-Agglo procedure provide a reasonable approximation to the true densities.

## 6. Real data

In this section we analyze the results obtained by applying the E-Divisive and E-Aggllo methods to two real datasets. We first apply our procedures to the micro-array aCGH data from [Bleakley and Vert \(2011\)](#). In this dataset we are provided with records of the copy-number variations for multiple individuals. Next we examine a set of financial time series. For this we consider weekly log returns of the companies which compose the Dow Jones Industrial Average.

### *Micro-array data*

This dataset consists of micro-array data for 57 different individuals with a bladder tumor. Since all individuals have the same disease, we would expect the change point locations to be almost identical on each micro-array set. In this setting, a change point would correspond to a change in copy-number, which is assumed to be constant within each segment. The group fused lasso (GFL) approach taken by [Bleakley and Vert \(2011\)](#) is well suited for this task since it is designed to detect changes in mean. We compare the results of our E-Divisive and E-Aggllo approaches, when using  $\alpha = 2$ , to those obtained by the GFL. In addition, we also consider another nonparametric change point procedure which is able to detect changes in both mean and variability, called MultiRank ([Lung-Yut-Fong, Lévy-Leduc, and Cappé 2011](#)). The original dataset from [Bleakley and Vert \(2011\)](#) contained missing values, and thus our procedure could not be directly applied. Therefore, we removed all individuals for which more

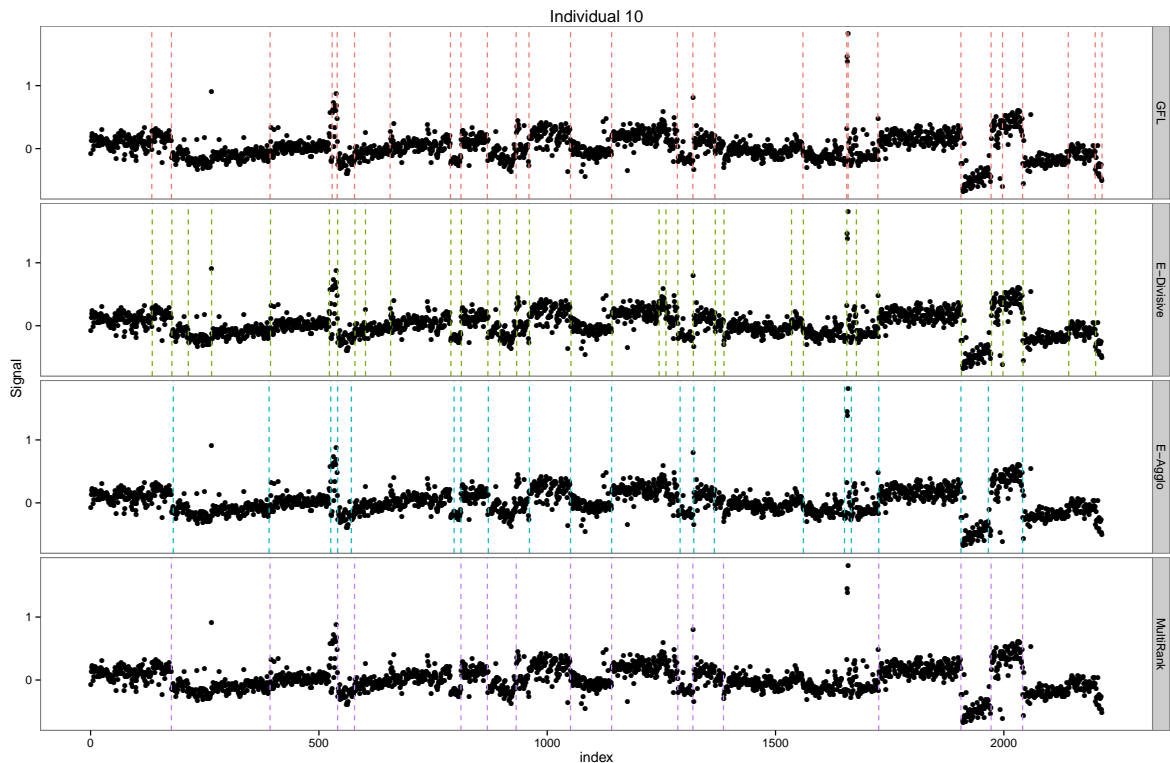


Figure 6: The aCGH data for individual 10. Estimated change point locations are indicated by dashed vertical lines.

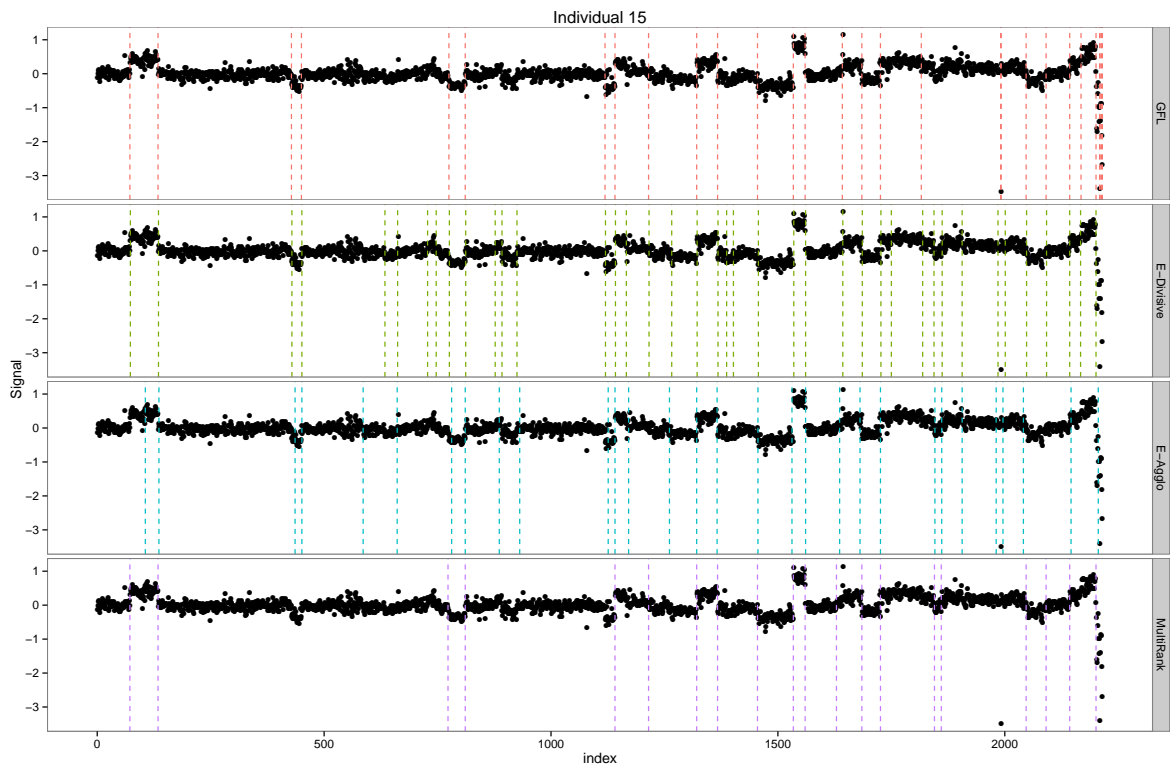


Figure 7: The aCGH data for individual 15. Estimated change point locations are indicated by dashed vertical lines.

than 7% of the values were missing. The remaining missing values we replaced by the average of their neighboring values. After performing this cleaning process, we were left with a sample of  $d = 43$  individuals and size  $T = 2215$ . This dataset can be obtained through the following R commands;

```
R> data("ACGH", package = "ecp")
R> acghData <- ACGH$data
```

When applied to the full 43 dimensional series, the GFL procedure estimated 14 change points and the MultiRank procedure estimated 43. When using  $\alpha = 2$ , the E-Divisive procedure estimated 86 change points and the E-Aggllo procedure estimated 28.

Figures 6 and 7 provide the results of applying the various methods to a subsample of two individuals (persons 10 and 15). The E-Divisive procedure was run with `min.size = 15`, and `R = 499`, and the initial segmentation provided to the E-Aggllo method consisted of equally sized segments of length 15. The marginal series are plotted, and the dashed lines are the estimated change point locations.

Looking at the returned estimated change point locations for the full 43-dimensional series we notice that both the E-Divisive and E-Aggllo methods identified all of the change points returned by the GFL procedure. Further examination also shows that in addition to those change points found by the GFL procedure, the E-Divisive procedure also identified changes in the means of the marginal series. However, if we examine the first 14 to 20 change points

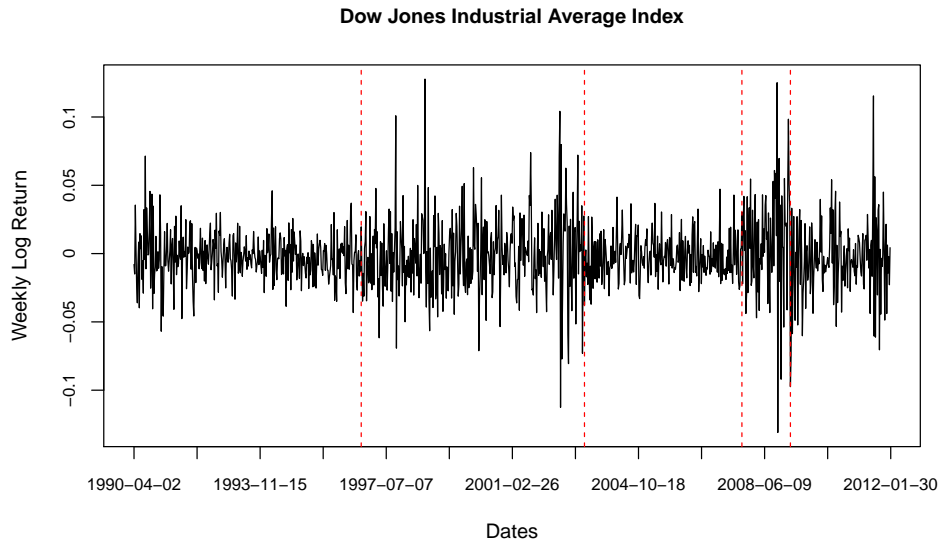


Figure 8: Weekly log returns for the Dow Jones Industrial Average index from April 1990 to January 2012. The dashed vertical lines indicate the locations of estimated change points. The estimated change points are located at 1996-10-21, 2003-03-31, 2007-10-15, and 2009-03-09.

estimated by the E-Divisive procedure we observe that they are those obtained by the GFL approach. This phenomenon however, does not appear when looking at the results from the E-Aggllo procedure. Intuitively this is due to the fact that we must provide an initial segmentation of the series, which places stronger limitations on possible change point locations, than does specifying a minimum segment size.

### *Financial data*

Next we consider weekly log returns for the companies which compose the Dow Jones Industrial Average (DJIA). The time period under consideration is April 1990 to January 2012, thus providing us with  $T = 1139$  observations. Since the time series for Kraft Foods Inc. does not span this entire period, it is not included in our analysis. This dataset is accessible by running `data("DJIA", package = "ecp")`.

When applied to the 29 dimensional series, the E-Divisive method identified change points at 1998-07-13, 2003-03-24, 2008-09-15, and 2009-05-11. The change points at 2009-05-11 and 2008-09-15 correspond to the release of the Supervisory Capital Asset Management program results, and the Lehman Brothers bankruptcy filing, respectively. If we initially segment the dataset into segments of length 30 and apply the E-Aggllo procedure, we identify change points at 2000-01-30, 2002-11-18, 2008-08-18, and 2009-03-16. The change points at 2000-01-03 and 2009-03-16 correspond to the passing of the Gramm-Leach-Bliley Act and the American Recovery and Reinvestment Act respectively.

For comparison we also considered the univariate time series for the DJIA index weekly log returns. In this setting, the E-Divisive method identified change points at 1996-10-21, 2003-03-31, 2007-10-15, and 2009-03-09. While the E-Aggllo method identified change points at 2008-08-18 and 2009-03-16. Once again, some of these change points correspond to major financial events. The change point at 2009-03-09 correspond to Moody's rating agency threatening



to downgrade Wells Fargo & Co., JP Morgan Chase & Co., and Bank of America Corp. The 2007-10-15 change point is located around the time of the financial meltdown caused by subprime mortgages. In both the univariate and multivariate cases the change point in March 2003 is around the time of the 2003 U.S. invasion of Iraq. A plot of the DJIA weekly log returns is provided in Figure 8 along with the locations of the estimated change points by the E-Divisive method.

For the E-Divisive method, the set of change points obtained from the univariate and multivariate analysis closely corresponds to the same events. However, in the case of the E-Aggl method, the multivariate analysis is able to identify significant events that were not able to be detected from the univariate series. For this reason, we would argue that regardless of the method being used, it is recommended that multivariate analysis be performed.

## 7. Performance analysis

To compare the performance of different change point methods we used the Rand index (Rand 1971) as well as Morey and Agresti’s adjusted Rand index (Morey and Agresti 1984). These indices provide a measure of similarity between two different segmentations of the same set of observations.

The Rand index evaluates similarity by examining the segment membership of *pairs* of observations. A shortcoming of the Rand index is that it does not measure departure from a given baseline model, thus making it difficult to compare two different estimated segmentations. The hypergeometric model is a popular choice for the baseline, and is used by Hubert and Arabie (1985) and Fowlkes and Mallows (1983).

In our simulation study the Rand and adjusted Rand Indices are determined by comparing the segmentation created by a change point procedure and the true segmentation. We compare the performance of our E-Divisive procedure against that of our E-Aggl. The results of the simulations are provided in Tables 1, 2 and 3. Tables 1 and 2 provide the results for

		Change in mean		Change in variance		
$T$	$\mu$	E-Divisive	E-Aggl	$\sigma$	E-Divisive	E-Aggl
150	1	0.939 <sub>0.003</sub>	0.834 <sub>0.006</sub>	2	0.507 <sub>0.008</sub>	0.580 <sub>0.003</sub>
	2	0.992 <sub>4.7×10<sup>-4</sup></sub>	0.990 <sub>0.001</sub>	5	0.971 <sub>0.001</sub>	0.935 <sub>0.003</sub>
	4	1.000 <sub>3.7×10<sup>-5</sup></sub>	1.000 <sub>0.000</sub>	10	0.985 <sub>8.6×10<sup>-4</sup></sub>	0.975 <sub>0.002</sub>
300	1	0.969 <sub>9.6×10<sup>-4</sup></sub>	0.899 <sub>0.005</sub>	2	0.728 <sub>0.009</sub>	0.505 <sub>0.002</sub>
	2	0.994 <sub>3.9×10<sup>-4</sup></sub>	0.994 <sub>6.4×10<sup>-4</sup></sub>	5	0.987 <sub>6.1×10<sup>-4</sup></sub>	0.966 <sub>0.002</sub>
	4	0.998 <sub>3.3×10<sup>-4</sup></sub>	1.000 <sub>0.000</sub>	10	0.992 <sub>5.1×10<sup>-4</sup></sub>	0.988 <sub>9.2×10<sup>-4</sup></sub>
600	1	0.985 <sub>5.4×10<sup>-4</sup></sub>	0.961 <sub>0.003</sub>	2	0.954 <sub>0.003</sub>	0.471 <sub>1.8×10<sup>-4</sup></sub>
	2	0.996 <sub>3.8×10<sup>-4</sup></sub>	0.997 <sub>3.1×10<sup>-4</sup></sub>	5	0.993 <sub>4.1×10<sup>-4</sup></sub>	0.982 <sub>0.001</sub>
	4	0.997 <sub>3.9×10<sup>-4</sup></sub>	1.000 <sub>0.000</sub>	10	0.995 <sub>3.8×10<sup>-4</sup></sub>	0.992 <sub>5.5×10<sup>-4</sup></sub>

Table 1: Average Rand index and standard errors from 1,000 simulations for the E-Divisive and E-Aggl methods. Each sample has  $T = 150, 300$  or  $600$  observations, consisting of three equally sized clusters, with distributions  $N(0, 1), G, N(0, 1)$ , respectively. For changes in mean  $G \equiv N(\mu, 1)$ , with  $\mu = 1, 2$ , and  $4$ ; for changes in variance  $G \equiv N(0, \sigma^2)$ , with  $\sigma = 2, 5$ , and  $10$ .

Change in tail			
$T$	$\nu$	E-Divisive	E-Agglo
150	16	0.348 <sub>0.003</sub>	0.534 <sub>9.5×10<sup>-4</sup></sub>
	8	0.347 <sub>0.003</sub>	0.535 <sub>9.8×10<sup>-4</sup></sub>
	2	0.365 <sub>0.004</sub>	0.540 <sub>1.2×10<sup>-3</sup></sub>
300	16	0.347 <sub>0.003</sub>	0.492 <sub>3.5×10<sup>-4</sup></sub>
	8	0.350 <sub>0.003</sub>	0.492 <sub>3.4×10<sup>-4</sup></sub>
	2	0.399 <sub>0.005</sub>	0.494 <sub>4.5×10<sup>-4</sup></sub>
600	16	0.349 <sub>0.003</sub>	0.469 <sub>1.4×10<sup>-4</sup></sub>
	8	0.347 <sub>0.002</sub>	0.469 <sub>1.4×10<sup>-4</sup></sub>
	2	0.431 <sub>0.006</sub>	0.469 <sub>1.5×10<sup>-4</sup></sub>

Table 2: Average Rand index and standard errors from 1,000 simulations for the E-Divisive and E-Agglo methods. Each sample has  $T = 150, 300$  or  $600$  observations, consisting of three equally sized clusters, with distributions  $N(0, 1), G, N(0, 1)$ , respectively. For the changes in tail shape  $G \equiv t_\nu(0, 1)$ , with  $\nu = 16, 8$ , and  $2$ .

		Change in mean		Change in correlation		
$T$	$\mu$	E-Divisive	E-Agglo	$\rho$	E-Divisive	E-Agglo
300	1	0.986 <sub>5.7×10<sup>-4</sup></sub>	0.976 <sub>0.002</sub>	0.5	0.351 <sub>0.003</sub>	0.482 <sub>1.8×10<sup>-4</sup></sub>
	2	0.997 <sub>4.0×10<sup>-4</sup></sub>	0.999 <sub>2.2×10<sup>-4</sup></sub>	0.7	0.349 <sub>0.003</sub>	0.484 <sub>2.0×10<sup>-4</sup></sub>
	3	0.996 <sub>4.7×10<sup>-4</sup></sub>	1.000 <sub>6.5×10<sup>-5</sup></sub>	0.9	0.362 <sub>0.003</sub>	0.487 <sub>3.0×10<sup>-4</sup></sub>
600	1	0.992 <sub>4.3×10<sup>-4</sup></sub>	0.985 <sub>0.001</sub>	0.5	0.351 <sub>0.002</sub>	0.464 <sub>7.1×10<sup>-5</sup></sub>
	2	0.998 <sub>3.4×10<sup>-4</sup></sub>	0.999 <sub>1.5×10<sup>-4</sup></sub>	0.7	0.358 <sub>0.003</sub>	0.464 <sub>7.8×10<sup>-5</sup></sub>
	3	0.997 <sub>4.0×10<sup>-4</sup></sub>	1.000 <sub>0.000</sub>	0.9	0.413 <sub>0.006</sub>	0.466 <sub>9.5×10<sup>-5</sup></sub>
900	1	0.994 <sub>3.6×10<sup>-4</sup></sub>	0.991 <sub>5.4×10<sup>-4</sup></sub>	0.5	0.356 <sub>0.003</sub>	0.457 <sub>3.7×10<sup>-5</sup></sub>
	2	0.997 <sub>3.4×10<sup>-4</sup></sub>	1.000 <sub>9.9×10<sup>-5</sup></sub>	0.7	0.373 <sub>0.004</sub>	0.458 <sub>4.2×10<sup>-5</sup></sub>
	3	0.998 <sub>3.1×10<sup>-4</sup></sub>	1.000 <sub>0.000</sub>	0.9	0.719 <sub>0.009</sub>	0.459 <sub>5.7×10<sup>-5</sup></sub>

Table 3: Average Rand index and standard errors from 1,000 simulations for the E-Divisive and E-Agglo methods, when applied to multivariate time series with  $d = 2$ . Each sample has  $T = 150, 300$  or  $600$  observations, consisting of three equally sized clusters, with distributions  $N_2(0, I), G, N_2(0, I)$ , respectively. For changes in mean  $G \equiv N_2(\mu, I)$ , with  $\mu = (1, 1)^\top, (2, 2)^\top$ , and  $(3, 3)^\top$ ; for changes in correlation  $G \equiv N(0, \Sigma_\rho)$ , in which the diagonal elements of  $\Sigma_\rho$  are 1 and the off-diagonal are  $\rho$ , with  $\rho = 0.5, 0.7$ , and  $0.9$ .

simulations with univariate time series, while Table 3 provides the results for the multivariate time series. In these tables, average Rand index along with standard errors are reported for 1000 simulations. Although not reported, similar results are obtained for the average adjusted Rand index.

Both the Rand index and adjusted Rand index can be easily obtained through the use of the `adjustedRand` function in the `clues` package (Chang, Qiu, Zamar, Lazarus, and Wang 2010). If  $U$  and  $V$  are membership vectors for two different segmentations of the data, then the required index values are obtained as follows,

```
R> library("clues")
R> RAND <- adjustedRand(U, V)
```

The Rand index is stored in `RAND[1]`, while `RAND[2]` and `RAND[3]` store various adjusted Rand indices. These adjusted Rand indices make different assumptions on the baseline model, and thus arrive at different values for the expected Rand index.

## 8. Conclusion

The `ecp` package is able to perform nonparametric change point analysis of multivariate data. The package provides two primary methods for performing analysis, each of which is able to determine the number of change points without user input. The only necessary user-provided parameter, apart from the data itself, is the choice of  $\alpha$ . If  $\alpha$  is selected to lie in the interval  $(0, 2)$ , then the methods provided by this package are able to detect *any* type of distributional change within the observed series, provided that the absolute  $\alpha$ th moments exist.

The E-Divisive method sequentially tests the statistical significance of each change point estimate given the previously estimated change locations, while the E-Agglo method proceeds by optimizing a goodness-of-fit statistic. For this reason, we prefer to use the E-Divisive method, even though its running time is output-sensitive and depends on the number of estimated change points.

Through the provided examples, applications to real data, and simulations (Matteson and James 2014), we observe that the E-Divisive approach obtains reasonable estimates for the locations of change points. Currently both the E-Divisive and E-Agglo methods have running times that are quadratic relative to the size of the time series. Future version of this package will attempt to reduce this to a linear relationship, or provide methods that can be used to quickly provide approximations.

## References

- Akoglu L, Faloutsos C (2010). “Event Detection in Time Series of Mobile Communication Graphs.” In *Proceedings of Army Science Conference*. 2010 Army Science Conference.
- Bleakley K, Vert JP (2011). “The Group Fused Lasso for Multiple Change-Point Detection.” *Technical Report HAL-00602121*, Bioinformatics Center (CBIO).
- Bolton R, Hand D (2002). “Statistical Fraud Detection: A Review.” *Statistical Science*, **17**(3), 235–255.
- Chang F, Qiu W, Zamar RH, Lazarus R, Wang X (2010). “`clues`: An R Package for Nonparametric Clustering Based on Local Shrinking.” *Journal of Statistical Software*, **33**(4), 1–16. URL <http://www.jstatsoft.org/v33/i04/>.
- Chasalow S (2012). *combinat: Combinatorics Utilities*. R package version 0.0-8, URL <http://CRAN.R-project.org/package=combinat>.
- Erdman C, Emerson JW (2007). “`bcp`: An R Package for Performing a Bayesian Analysis of Change Point Problems.” *Journal of Statistical Software*, **23**(3), 1–13. URL <http://www.jstatsoft.org/v23/i03/>.

- Fowlkes EB, Mallows CL (1983). “A Method for Comparing Two Hierarchical Clusterings.” *Journal of the American Statistical Association*, **78**(383), 553–569.
- Gandy A (2009). “Sequential Implementation of Monte Carlo Tests with Uniformly Bounded Resampling Risk.” *Journal of the American Statistical Association*, **104**(488), 1504–1511.
- Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2014). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-0, URL <http://CRAN.R-project.org/package=mvtnorm>.
- Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A (2007). “A Kernel Method for the Two Sample Problem.” In *Advances in Neural Information Processing Systems 19*, pp. 513–520. MIT Press.
- Hoeffding W (1961). “The Strong Law of Large Numbers for U-Statistics.” *Technical Report 302*, North Carolina State University, Department of Statistics.
- Hubert L, Arabie P (1985). “Comparing Partitions.” *Journal of Classification*, **2**(1), 193–218.
- Killick R, Eckley IA (2014). “**changepoint**: An R Package for Changepoint Analysis.” *Journal of Statistical Software*, **58**(3), 1–19. URL <http://www.jstatsoft.org/v58/i03/>.
- Killick R, Fearnhead P, Eckley IA (2012). “Optimal Detection of Changepoints with a Linear Computational Cost.” *Journal of the American Statistical Association*, **107**(500), 1590–1598.
- Lung-Yut-Fong A, Lévy-Leduc C, Cappé O (2011). “Homogeneity and Change-Point Detection Tests for Multivariate Data Using Rank Statistics.” arXiv:1107.1971 [math.ST], URL <http://arxiv.org/abs/1107.1971>.
- Mampaey M, Vreeken J (2013). “Summarizing Categorical Data by Clustering Attributes.” *Data Mining and Knowledge Discovery*, **26**(1), 130–173.
- Matteson DS, James NA (2014). “A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data.” *Journal of the American Statistical Association*, **109**(505), 334–345.
- Morey LC, Agresti A (1984). “The Measurement of Classification Agreement: An Adjustment to the Rand Statistic for Chance Agreement.” *Educational and Psychological Measurement*, **44**(1), 33–37.
- Muggeo VMR, Adelfio G (2011). “Efficient Change Point Detection for Genomic Sequences of Continuous Measurements.” *Bioinformatics*, **27**(2), 161–166.
- Rand WM (1971). “Objective Criteria for the Evaluation of Clustering Methods.” *Journal of the American Statistical Association*, **66**(336), 846–850.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Ross GJ (2013). *cpm: Sequential Parametric and Nonparametric Change Detection*. R package version 1.1, URL <http://CRAN.R-project.org/package=cpm>.

- Schauer K, Duong T, Bleakley K, Bardin S, Bornens M, Goud B (2010). “Probabilistic Density Maps to Study Global Endomembrane Organization.” *Nature Methods*, **7**(7), 560–566.
- Sequeira K, Zaki M (2002). “ADMIT: Anomaly-Based Data Mining for Intrusions.” In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02. ACM.
- Székely GJ, Rizzo ML (2005). “Hierarchical Clustering Via Joint Between-Within Distances: Extending Ward’s Minimum Variance Method.” *Journal of Classification*, **22**(2), 151–183.
- Székely GJ, Rizzo ML (2010). “Disco Analysis: A Nonparametric Extension of Analysis of Variance.” *The Annals of Applied Statistics*, **4**(2), 1034–1055.
- Talih M, Hengartner N (2005). “Structural Learning with Time-Varying Components: Tracking the Cross-Section of Financial Time Series.” *Journal of the Royal Statistical Society B*, **67**(3), 321–341.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. URL <http://www.stats.ox.ac.uk/pub/MASS4>.
- Zeileis A, Kleiber C, Krämer W, Hornik K (2003). “Testing and Dating of Structural Changes in Practice.” *Computational Statistics & Data Analysis*, **44**(1–2), 109–123.
- Zeileis A, Leisch F, Hornik K, Kleiber C (2002). “**strucchange**: An R Package for Testing for Structural Change in Linear Regression Models.” *Journal of Statistical Software*, **7**(2), 1–38. URL <http://www.jstatsoft.org/v07/i02/>.

## A. Additional details

This appendix provides additional details about the implementation of both the E-Divisive and E-Agglo methods in the **ecp** package.

### A.1. Divisive outline

The E-Divisive method estimates change points with a bisection approach. In Algorithms 1 and 2, segment  $C_i$  contains all observations in time interval  $[\ell_i, r_i)$ . Algorithm 2 demonstrates the procedure used to identify a single change point. The computational time to maximize over  $(\tau, \kappa)$  is reduced to  $\mathcal{O}(T^2)$  by using memoization. Memoization also allows Algorithm 2 to execute its for loop at most twice. The permutation test is outlined by Algorithm 3. When given the segmentation  $C$ , a permutation is only allowed to reorder observations so that they remain within their original segments.

### A.2. Agglomerative outline

The E-Agglo method estimates change point by maximizing the goodness-of-fit statistic given by Equation 3. The method must be provided an initial segmentation of the series. Segments are then merged in order to maximize the goodness-of-fit statistic. As segments are merged, their between-within distances also need to be updated. The following result due to Székely and Rizzo (2005) greatly reduces the computational time necessary to perform these updates.

**Lemma 3.** *Suppose that  $C_1, C_2$ , and  $C_3$  are disjoint segments with respective sizes  $m_1, m_2$ , and  $m_3$ . Then if  $C_1$  and  $C_2$  are merged to form the segment  $C_1 \cup C_2$ ,*

$$\widehat{\mathcal{E}}(C_1 \cup C_2, C_3; \alpha) = \frac{m_1 + m_3}{m_1 + m_2 + m_3} \widehat{\mathcal{E}}(C_1, C_3; \alpha) + \frac{m_2 + m_3}{m_1 + m_2 + m_3} \widehat{\mathcal{E}}(C_2, C_3; \alpha) - \frac{m_3}{m_1 + m_2 + m_3} \widehat{\mathcal{E}}(C_1, C_2; \alpha).$$

Algorithm 4 is an outline for the agglomerative procedure. In this outline  $C_{i+k}$  ( $C_{i-k}$ ) is the segment that is  $k$  segments to the right (left) of  $C_i$ .

---

**Algorithm 1:** Outline of the divisive procedure.

---

**Inputs :** Time series  $Z$ , significance level  $p_0$ , minimum segment size  $m$ , the maximum number of permutations for the permutation test  $R$ , the uniform resampling error bound  $eps$ , epsilon spending rate  $h$ , and  $\alpha \in (0, 2]$ .

**Output:** A segmentation of the time series.

Create distance matrix  $Z_{ij}^\alpha = |Z_i - Z_j|^\alpha$

**while** *Have not found a statistically insignificant change point*

    | Estimate next most likely change point location

    | Test estimated change point for statistical significance

    | **if** *Change point is statistically significant* **then**

        | Update the segmentation

    | **end**

**endwhile**

**return** Final segmentation

---

---

**Algorithm 2:** Outline of procedure to locate a single change point.

---

**Inputs :** Segmentation  $C$ , distance matrix  $D$ , minimum segment size  $m$ .

**Output:** A triple  $(x, y, z)$  containing the following information: a segment identifier, a distance within a segment, a weighted sample divergence.

best =  $-\infty$

loc = 0

**for** Segments  $C_i \in C$

$A =$  Within distance for  $[\ell_i, \ell_i + m)$

**for**  $\kappa \in \{\ell_i + m + 2, \dots, r_i + 1\}$

        Calculate and store between and within distances for current choice of  $\kappa$

        Calculate test statistic

**if** *Test statistic*  $\geq$  *best* **then**

            Update best

            Update loc to  $m$

**end**

**endfor**

**for**  $\tau \in \{\ell_i + m + 1, \dots, r_i - m\}$

        Update within distance for left segment

**for**  $\kappa \in \{\tau + m + 1, \dots, r_i + 1\}$

            Update remaining between and within distances for current choice of  $\kappa$

            Calculate test statistic

**if** *Test statistic*  $\geq$  *best* **then**

                Update best

                Update loc to  $\tau$

**end**

**endfor**

**endfor**

**endfor**

**return** Which segment to divide, loc, and best

---

---

**Algorithm 3:** Outline of the permutation test.

---

**Inputs :** Distance matrix  $D$ , observed test statistic  $\nu$ , maximum number of permutations  $R$ , uniform resampling error bound  $eps$ , epsilon spending rate  $h$ , segmentation  $C$ , minimum segment size  $m$ .

**Output:** An approximate  $p$  value.

```

over = 1
for i ∈ {1, 2, ..., R}
  | Permute rows and columns of  $D$  based on the segmentation  $C$  to create  $D'$ 
  | Obtain test statistic for permuted observations
  | if Permuted test statistic ≥ observed test statistic then
  |   | over = over + 1
  | end
  | if An early termination condition is satisfied then
  |   | return over/(i+1)
  | end
endfor
return over/(R+1)

```

---



---

**Algorithm 4:** Outline of the agglomerative procedure.

---

**Inputs :** An initial segmentation  $C$ , a time series  $Z$ , a penalty function  $f(\vec{\tau})$ , and  $\alpha \in (0, 2]$ .

**Output:** A segmentation of the time series.

$N$  = number of time series observations

Create distance matrix  $D_{i,j} = \widehat{\mathcal{E}}(C_i, C_j; \alpha)$

Obtain initial penalized goodness-of-fit (gof) statistic

**for**  $K \in \{N, N + 1, \dots, 2N - 3\}$

    | Merge best candidate segments

    | Update current gof

    | **if** *Current gof* ≥ *largest gof so far* **then**

        | Update largest gof

    | **end**

**endfor**

Penalize the sequence of obtained gof statistics

Choose best segmentation based on penalized gof statistics

**return** Best segmentation

---



**Affiliation:**

Nicholas A. James  
School of Operations Research and Information Engineering  
Cornell University  
206 Rhodes Hall  
Ithaca NY 14853, United States of America  
E-mail: [nj89@cornell.edu](mailto:nj89@cornell.edu)  
URL: <https://courses.cit.cornell.edu/nj89/>

David S. Matteson  
Department of Statistical Science  
Cornell University  
1196 Comstock Hall  
Ithaca NY 14853, United States of America  
E-mail: [matteson@cornell.edu](mailto:matteson@cornell.edu)  
URL: <http://www.stat.cornell.edu/~matteson/>