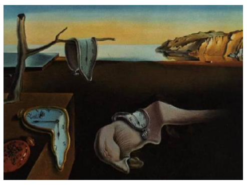

Journal of Statistical Software
December 2007, Volume 23, Issue 3. http://www.jstatsoft.org/

bcp: An R Package for Performing a Bayesian Analysis of Change Point Problems

Chandra Erdman
Yale University

John W. Emerson
Yale University

Abstract

Barry and Hartigan (1993) propose a Bayesian analysis for change point problems. We provide a brief summary of selected work on change point problems, both preceding and following Barry and Hartigan. We outline Barry and Hartigan's approach and offer a new R package, pkgbcp (Erdman and Emerson 2007), implementing their analysis. We discuss two frequentist alternatives to the Bayesian analysis, the recursive circular binary segmentation algorithm (Olshen and Venkatraman 2004) and the dynamic programming algorithm of (Bai and Perron 2003). We illustrate the application of **bcp** with economic and microarray data from the literature.

Keywords: change point, structural change, product partition model.

1. Introduction

During the Great Depression, farmers were interested in estimating spatial changes in insect populations threatening crops – a change point problem. Although they discovered that the insect populations didn't change significantly within fields (Finney 1946), change point methods have since been applied to problems in economics, gynecology, and survival analysis, for example. More recently, certain types of microarray data are suitable for change point analysis.

Circular binary segmentation (CBS, Olshen and Venkatraman 2004), a modification of binary segmentation (Sen and Srivastava 1975), is a popular, recursive change point algorithm. Although it was designed for the analysis of microarray data, it is generally applicable to any change point problem. Both procedures assume normality, estimating locations of change points using a likelihood ratio statistic. The R implementation of CBS is available on Bioconductor (Gentleman *et al.* 2004). Another popular algorithm, breakpoints (BP, Bai and Perron 2003), is available in the R package **strucchange** (Zeileis *et al.* 2002, 2003). For a given number of change points, `breakpoints()` uses least squares regression to estimate the locations of the changes. The function then selects an optimal model (choosing the number of change points) using the Bayesian information criterion (BIC) by default. We implement a Bayesian alternative to these procedures in the R package **bcp** – the approach of Barry and Hartigan (1993) based on a product partition model.

Section 2 summarizes the work of Bai and Perron (2003), Olshen and Venkatraman (2004), and Barry and Hartigan (1993), and describes a numerical challenge posed by the implementation of Barry and Hartigan's procedure (BH). Section 3 describes the **bcp** package and illustrates its usage in a simulation study and applied to microarray and economic data. We conclude by discussing the strengths and weaknesses of these three procedures and directions for future work.

2. Methods

This paper offers a new R implementation of the Bayesian change point procedure proposed by Barry and Hartigan (1993). While frequentist procedures for change point analysis estimate specific locations of change points, the Bayesian procedure offers a probability distribution – the probability of a change point at each location in a sequence. To assist the reader, we begin by discussing two frequentist alternatives, of Bai and Perron (2003), and Olshen and Venkatraman (2004). We then describe the Bayesian approach and address a numerical challenge proposed by the implementation of BH. We use the notation of the authors whenever possible.

2.1. Bai and Perron's dynamic programming algorithm

Bai and Perron's method uses a dynamic programming algorithm to identify optimal partitions with varying numbers of segments. They provide options for both the minimum segment length, `h`, and the maximum number of breaks, `m`. For each possible number of breaks $k \leq$ `m`, they obtain the optimal break point locations by minimizing the within-segment sums of squares. The R implementation of BP reports the partition achieving the lowest BIC by default; users may also compute the log likelihood and Akaike information criterion (AIC) of partitions using the `logLik()` and `AIC()` functions.

We use Bai and Perron's pure structural change model for simple change point examples and simulations in Section 3.3. Readers interested in the more general framework of Bai and Perron's method (a structural change model with covariates) are encouraged to see Bai and Perron (2003).

2.2. Recursive binary and circular binary segmentation

Both binary segmentation and circular binary segmentation consider observations, $X_1, ..., X_n$, ordered in time or space. Let $S_i = X_1 + \cdots + X_i$, for $1 \leq i \leq n$, denote the partial sums of the observations. By assuming the data are assumed normally distributed with a known variance, binary segmentation uses the likelihood ratio statistic for testing the null hypothesis of no change point against the alternative of exactly one change point at an unknown location i (Sen and Srivastava 1975). The likelihood ratio statistic is given by $Z_B = max|Z_i|$, where

$$Z_i = 1/i + 1/(n-i)^{1/2} S_i/i - (S_n - S_i)/(n-i), \tag{1}$$

for $1 \leq i < n$. The null hypothesis is rejected if this statistic exceeds the upper α^{th} quantile of the null distribution of Z_B, in which case the location of the change point is estimated as that i for which $Z_B = |Z_i|$ (Olshen and Venkatraman 2004). The test is applied recursively until no changes are detected in any of the segments of the partition.

Because the binary segmentation algorithm is based on a test to detect a single change, it may have trouble detecting a small segment buried in the middle of a large segment (Olshen and Venkatraman 2004). Olshen and Venkatraman proposed the circular binary segmentation algorithm to address this problem. Using CBS, each segment under consideration is connected at the two ends, forming a circle. The likelihood ratio statistic for testing the hypothesis that the arc extending from $i+1$ to j and its complement have different means is given by

$$Z_{ij} = \frac{(S_j - S_i)/(j-i) - (S_n - S_j + S_i)/(n-j+i)}{[1/(j-i) + 1/(n-j+i)]^{1/2}}, \tag{2}$$

for $1 \leq i < j \leq n$. This modification of binary segmentation is based on the statistic $Z_C = max|Z_{ij}|$, for $1 \leq i < j \leq n$, and rejects the null hypothesis if this statistic exceeds an appropriate threshold based on the null distribution of Z_C. Note that this procedure allows either a single change $(j = n)$ or two changes $(j < n)$. If the null hypothesis is rejected, the change points are estimated to be i and j such that $Z_C = |Z_{ij}|$ and, again, the procedure is applied recursively to the resulting sub-segments until no additional changes are detected. Unlike BP, there is no guarantee that Binary or circular binary segmentation achieves the optimal change point locations for the recommended number of changes.

2.3. Barry and Hartigan

As in CBS, Barry and Hartigan assume that the observations are independent $N(\mu_i, \sigma^2)$, and that the probability of a change point at a position i is p, independently at each i. However, the assumption of independent observations could be weakened "because all that is required is that, given the partition and the parameters, observations in different blocks are mutually independent" (Barry and Hartigan 1993, p. 310). The prior distribution of μ_{ij} (the mean of the block beginning at position $i+1$ and ending at position j) is chosen as $N(\mu_0, \sigma_0^2/(j-i))$. This choice of prior allows "weak signals provided that there are sufficient data to estimate them" (Barry and Hartigan 1993, p. 311). Although an exact implementation of Barry and Hartigan's Bayes procedure is possible, the calculations are $O(n^3)$; we implement an MCMC approximation that is $O(n^2)$. The reader is encouraged to refer to Barry and Hartigan (1993) for the full theoretical framework, and our presentation will conform to their notation.

The algorithm uses a partition $\rho = (U_1, U_2, ..., U_n)$, where $U_i = 1$ indicates a change point at position $i+1$; we initialize U_i to 0 for all $i < n$, with $U_n \equiv 1$. In each step of the Markov chain, at each position i, a value of U_i is drawn from the conditional distribution of U_i given the data and the current partition. Following Barry and Hartigan, we let b denote the number of blocks obtained if $U_i = 0$, conditional on U_j, for $i \neq j$. The transition probability, p, for the conditional probability of a change point at the position $i+1$, may be obtained from the simplified ratio presented in Barry and Hartigan:

$$\frac{p_i}{1-p_i} = \frac{P(U_i = 1|\mathbf{X}, U_j, j \neq i)}{P(U_i = 0|\mathbf{X}, U_j, j \neq i)} \tag{3}$$

$$= \frac{[\int_0^\gamma p^b(1-p)^{n-b-1}\,dp]\left[\int_0^\lambda \frac{w^{b/2}}{(W_1+B_1w)^{(n-1)/2}}\,dw\right]}{[\int_0^\gamma p^{b-1}(1-p)^{n-b}\,dp]\left[\int_0^\lambda \frac{w^{(b-1)/2}}{(W_0+B_0w)^{(n-1)/2}}\,dw\right]} \tag{4}$$

where W_0, B_0, W_1 and B_1 are the within and between block sums of squares obtained when $U_i = 0$ and $U_i = 1$ respectively, and $\mathbf{X}$ is the data. The tuning parameters γ and λ may take values in $[0, 1]$, chosen so that this method "is effective in situations where there aren't too many changes (γ small), and where the changes that do occur are of a reasonable size (λ small)" (Barry and Hartigan 1993, p. 312). After each iteration, the posterior means are updated conditional on the current partition. A direct implementation of the BH MCMC algorithm is numerically unstable for long sequences because the integrands of

$$\int_0^\lambda \frac{w^{b/2}}{(W_1+B_1w)^{(n-1)/2}}\,dw$$

and

$$\int_0^\lambda \frac{w^{(b-1)/2}}{(W_0+B_0w)^{(n-1)/2}}\,dw$$

either diverge or go to 0 for long sequences. Fortunately, these integrals can be simplified as incomplete beta integrals. The odds of a change point at a particular position in the partition (given the data and the current partition) may be re-expressed as

$$\frac{p_i}{1-p_i} = \frac{P(U_i = 1|\mathbf{X}, U_j, j \neq i)}{P(U_i = 0|\mathbf{X}, U_j, j \neq i)}$$

$$= \left(\frac{W_0}{W_1}\right)^{\frac{n-b-2}{2}} \cdot \left(\frac{B_0}{B_1}\right)^{\frac{b+1}{2}} \cdot \sqrt{\frac{W_1}{B_1}} \cdot \frac{\int_0^{\frac{B_1\lambda/W_1}{1+B_1\lambda/W_1}} p^{(b+2)/2}(1-p)^{(n-b-3)/2}\,dp}{\int_0^{\frac{B_0\lambda/W_0}{1+B_0\lambda/W_0}} p^{(b+1)/2}(1-p)^{(n-b-2)/2}\,dp} \cdot \frac{\int_0^\gamma p^b(1-p)^{n-b-1}\,dp}{\int_0^\gamma p^{b-1}(1-p)^{n-b}\,dp}.$$

This expression consists of numerically stable terms, allowing application of the BH procedure to sequences of any length. The MCMC implementation of BH estimates the posterior distributions of the change points and the means, μ_{ij}.

3. Package bcp: Examples and simulations

Package **bcp** contains the main `bcp()` function, five methods (`summary()`, `print()`, `plot()`, `fitted()`, and `residuals()`), and two datasets. The function `bcp()` performs the BH analysis, taking six arguments:

- `x`: a numerical vector of data.
- `p0` and `w0`: optional values for Barry and Haritgan's hyperparameters γ and λ; these default to the value 0.2, which has been found to work well (see Yao (1984) and Barry and Hartigan (1993)).

- `burnin`: optional number of "burn-in" iterations that are excluded from the estimation of the posterior means and probabilities of changes. The chain settles very quickly in practice, and the default is 50.
- `mcmc`: optional number of iterations used in the estimation of the posterior means; the default is 500.
- `return.mcmc`: if `TRUE`, returns the partition and the associated conditional posterior means for each iteration; the default is set to `FALSE` and returns a summary of the chain.

After completing the analysis, `bcp()` returns an object of class "`bcp`" containing the following components:

- `data`: a copy of the data.
- `mcmc.means`: if `return.mcmc=TRUE`, contains the posterior means conditional on the current partition at the end of every iteration, otherwise `mcmc.means` is `NA`.
- `mcmc.rhos`: if `return.mcmc=TRUE`, contains the partition after each iteration, otherwise `mcmc.rhos` is `NA`.
- `blocks`: a vector of the number of blocks after each iteration.
- `posterior.mean`: a vector containing the posterior means.
- `posterior.var`: a vector containing the "naive" posterior variance for each postition, over the `mcmc` iterations.
- `posterior.prob`: a vector containing the posterior probability of a change point at each position.
- `p0`, `w0`, `burnin`, `mcmc` and `return.mcmc`: contain the specified values.

Package **bcp** contains five methods and two datasets::

- Methods:
  - The `plot()` method provides two plots summarizing the analysis. The first figure, "Posterior Means", displays the data along with the posterior mean of each position. The second figure, "Posterior Probability of a Change", shows the proportion of iterations resulting in a change point at each position.
  - The `summary()` and `print()` methods, modeled after the the `summary()` method for "`mcmc`" objects (Plummer *et al.* 2007), print a table of the posterior probability of a change in mean, along with the posterior mean and standard deviation for each position. After removing the `burnin` iterations, the `mcmc.means` component of a "`bcp`" object may be converted into an "`mcmc`" object to view a full "`mcmc`" summary, and to perform convergence tests. An example of this conversion is given in the **bcp** package documentation (Erdman and Emerson 2007).
  - The `fitted()` method returns the vector of posterior means.

  - The `residuals()` method returns the data minus the posterior means.

- Data:
  - `Coriell`: two array Comparative Genomic Hybridization (CGH) studies of Coriell cell lines, also appears in the **DNAcopy** package (Venkatraman and Olshen 2006) that performs CBS, taken originally from Snijders *et al.* (2001).
  - `RealInt`: US interest rate data, considered by Bai and Perron (2003) and Garcia and Perron (1996), also appears in the **strucchange** package that performs BP (Zeileis *et al.* 2002, 2003).

We consider these examples from the literature in Sections 3.1 and 3.2, and conclude with a simulation study mirroring the one presented in Barry and Hartigan (1993). In all cases, we allow the BP algorithm to consider all possible partitions (the default is a minimum segment length of 15), and we use the hyperparameters recommended by Barry and Hartigan; we do not "tune" any of the parameters for the examples or simulations.

3.1. Example: Coriell cell lines

The CGH studies of the Coriell cell lines were used by Venkatraman and Olshen (2006) to demonstrate CBS in **DNAcopy**, and by Fridlyand *et al.* (2004) to demonstrate their hidden markov model approach. CGH was developed as a method for detecting and mapping chromosomal aberrations in the genome (Kallioniemi *et al.* 1992). The procedure begins by dying tumor and normal tissue with different fluorochromes (usually red and green). If there have been no chromosomal aberrations in the tumor DNA, the mixture of the two samples will emit a yellow fluorescence. However, if there have been amplifications or deletions in the tumor sample, the mixture will emit a red or green fluorescence depending on the color with which the normal tissue was dyed. The fluorescence is then translated into DNA copy number.

Fridlyand *et al.* (2004) describes the Coriell copy number data as consisting "of 15 fibroblast cell lines containing cytogenetically mapped partial or whole-chromosome aneuploidy, and each array contained 2276 mapped BACs spotted in triplicate." UCSF SPOT software (Jain *et al.* 2002) was used to calculate the $\log_2$ratio ratios of the red-green signal intensities for each spot on the arrays, and to perform local background correction. A commercial program, SPROC, was used to map the data to its location in the genome, and as a filter to remove measurements based on a number of criteria including low reference/signal intensity (Snijders *et al.* 2001). Finally, the data were edited to remove measurements for which only one of the triplicates remained after the SPROC filter and/or the standard deviation of the triplicates was > 0.2 (Snijders *et al.* 2001).

When applied to chromosome 11 of `Coriell.05296`, for example, BP and CBS tend to agree on the location of the changes in copy number (and thus give identical copy number estimates). BH gives similar estimates when there are long blocks of roughly equal copy number, but gives very different estimates when there are shorter blocks and/or "outliers". Figure 1 shows that the BH estimates match the BP and CBS estimates well, except in the middle "block" where BH detects more activity. Short-lived changes, like those in the middle block, can be due to cross-hybridization, false signals (a speck on the array), or a true signal (a deletion perhaps on a single strand of the DNA). Because these clones were spotted in triplicate, a false signal is unlikely. The Bayesian approach clearly acknowledges this area of uncertainty, while the frequentist procedures only identify the two major breaks.

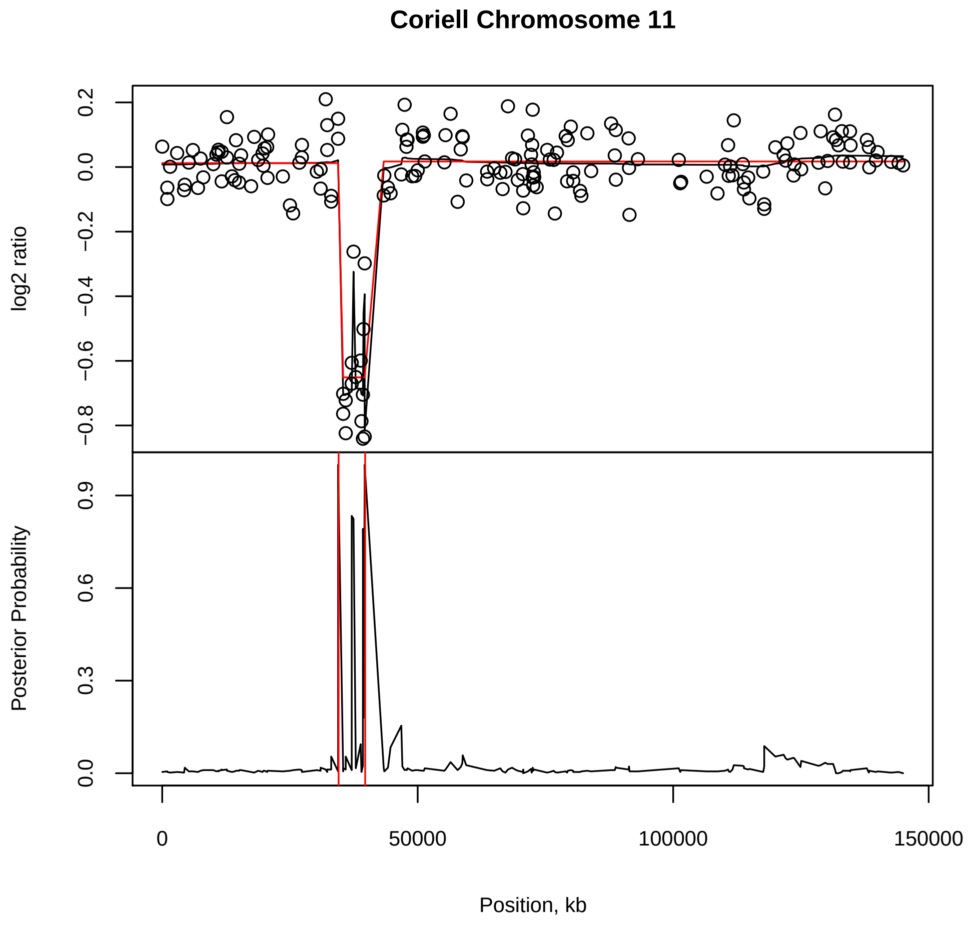

Figure 1: BH, BP, and CBS estimates on Coriell chromosome 11 (data from Snijders *et al.* 2001). The black lines represent BH, and the red lines represent CBS and BP. In the bottom plot, the black lines represent Barry and Hartigan's posterior probabilities of changes, and the red vertical lines represent the change point locations of CBS and BP.

```
R> data("coriell")
R> chrom11 <- as.vector(na.omit(coriell$Coriell.05296[coriell$Chromosome==11]))
R> n <- length(chrom11)
R> bcp.11 <- bcp(chrom11)
R> cbs <- segment(CNA(chrom11, rep(1, n), 1:n), verbose = 0)
R> cbs.11 <- rep(unlist(cbs$output[6]), unlist(cbs$output[5]))
R> bp.11 <- breakpoints(chrom11 ~ 1, h = 2)$breakpoints
```

3.2. Example: Interest rate time series

The `RealInt` dataset may be found in the **strucchange** package (Zeileis *et al.* 2002, 2003) and was examined in Bai and Perron (2003) and Garcia and Perron (1996). Garcia and Perron

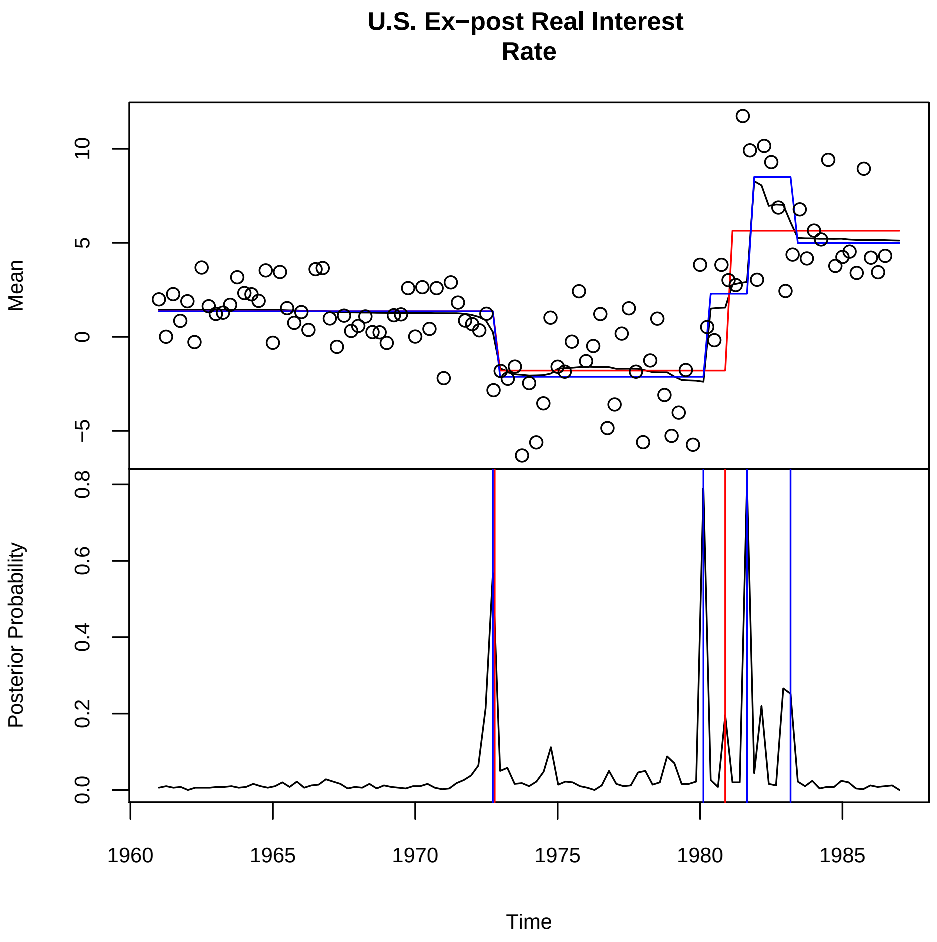

Figure 2: BH, BP, and CBS estimates on US ex-post real interest rate 1961–1986. The black lines represent BH, the blue lines represent BP, and the red lines represent CBS. In the bottom plot, the black lines represent Barry and Hartigan's posterior probabilities of changes, and the red and blue vertical lines represent the change point locations of CBS and BP respectively.

use the data "to assess if the ex-ante real interest rate is constant, at least over some long enough periods, or if it exhibits nonstationary behavior" (Garcia and Perron 1996, p. 111). `RealInt` contains 103 quarterly observations of the US ex-post real interest rate from 1961(1) to 1986(3). Figure 2 shows that BH and BP give similar estimates, detecting changes in 1972(3), 1979(4), 1981(2). Although BH is less sure of the change identified by BP around the fourth quarter of 1982, there is still a notable spike in the posterior probability of this change. CBS also identifies the first interest rate shift at the end of 1972, but splits BH and BP's third block (from 1979(4) - 1981(2)), giving a 3-block model.

Garcia and Perron note that "although some of the local optima seem to correspond to important economic events such as the change in the Federal Reserve operating procedures between the end of 1979 and 1982 or the rise in inflation in 1973, the global minimum does not have any ready economic interpretation" (Garcia and Perron 1996, p. 119). The authors use the Markov switching model of Hamilton (1989) and identify mean interest rate shifts in the beginning of 1973 and in mid-1981. Bai and Perron (2003) apply `breakpoints()` to `RealInt` with parameters `m`=5 and `h`=15 (i.e., they require at least 15 observations per segment, and restrict their search to partitions with 5 or less shifts). They apply a sequential *supF* test of k vs. $k+1$ breaks, and select the model with changes in 1966(4), 1972(3), and 1980(3). Because Bai and Perron tuned the parameters to suit their interpretation of the problem, their published result differs slightly from the one presented here.

```
R> data("RealInt")
R> n <- length(RealInt)
R> bcp.ri <- bcp(as.vector(RealInt), p0 = 0.1)
R> bp.ri <- breakpoints(RealInt ~ 1, h = 2)$breakpoints
R> cbs <- segment(CNA(RealInt, rep(1, n), 1:n), verbose = 0)
R> cbs.ri <- rep(unlist(cbs$output[6]), unlist(cbs$output[5]))
```

3.3. A simulation study

We consider the same 20 "scenes" used by Barry and Hartigan (a scene is a partition together with a set of means for the segments of the partition). For each scene, 60 observations are randomly drawn from one or more normal distributions with specified mean(s) and variance 1. We follow Barry and Hartigan's notation in describing the scenes. For example, the sequence 10^0 15^1 20^2 15^1 denotes a scene with four blocks of lengths 10, 15, 20, and 15 drawn from $N(0,1)$, $N(1,1)$, $N(2,1)$, and $N(1,1)$ respectively. For each of the 20 scenes, we simulate 1000 data sets, and apply CBS, BP, and BH to each of them. The differences in mean squared error and their standard errors are presented in Table 1. Specifically, for each data set, we calculate the mean squared errors for each method – the mean of the squared differences between the pointwise estimates and the true means. Thus, for each scene, we obtain 1000 MSEs for each method. Lastly, we calculate the mean and standard errors of the differences in these MSEs between methods, using BH as the baseline.

Table 1 shows that the BH algorithm outperforms CBS and BP in MSE in most of the scenes. These differences in MSE are especially pronounced when there are shorter blocks and/or outliers. For example, in Figure 3 scene 20, both CBS and BP fail to detect any of the regular, smaller changes, and BH performs significantly better than both methods. Also, in scene 18 with relatively large, short-lived changes, BH performs much better than CBS. In scenes 2 and 7, having two and three blocks respectively, CBS only does slightly better in MSE than BH, and BH does only slightly better than BP. In Figure 3 we see that all three methods do a reasonably good job at estimating the true block means for these scenes.

4. Conclusion

The current version, 1.8.4, of **bcp** (which uses R and C) is available for the R system for statistical computing (R Development Core Team 2007) from the Comprehensive R Archive Network at `http://CRAN.R-project.org/`. This implementation (performing the default 550 iterations) runs in approximately 0.75 seconds for a sequence of length 100 on a PC with Windows XP, a Pentium D Processor (2.99 GHz) and 3.50GB of RAM, compared with 0.06

	Scene	Mean(MSE(BH) - MSE(CBS))	Mean(MSE(BH) - MSE(BP))
1	60^0	$0.0305^{1.5}$	$0.0110^{1.7}$
2	$40^0\ 20^3$	$0.0088^{1.7}$	$-0.0106^{2.1}$
3	$40^0\ 20^2$	$0.0118^{2.4}$	$-0.0084^{2.3}$
4	$30^0\ 30^1$	$-0.0910^{2.9}$	$-0.0332^{2.3}$
5	$30^0\ 30^{0.5}$	$-0.0173^{1.4}$	$-0.0361^{1.6}$
6	$58^0\ 2^3$	$-0.1937^{2.6}$	$0.0108^{2.4}$
7	$15^0\ 30^2\ 15^0$	$0.0116^{2.6}$	$-0.0129^{2.7}$
8	$10^0\ 40^1\ 10^0$	$-0.0808^{1.9}$	$-0.0846^{2.1}$
9	$30^0\ 20^2\ 10^0$	$0.0208^{2.9}$	$-0.0114^{3.0}$
10	$4^0\ 1^5\ 55^0$	$-0.2550^{3.5}$	$-0.2292^{3.5}$
11	$10^1\ 15^0\ 20^2\ 15^1$	$-0.1213^{3.3}$	$-0.0923^{2.6}$
12	$10^1\ 10^2\ 10^1\ 30^0$	$-0.0899^{3.1}$	$-0.0756^{2.3}$
13	$25^2\ 1^0\ 14^5\ 20^3$	$-0.0481^{4.2}$	$-0.0217^{2.7}$
14	$15^3\ 5^0\ 5^5\ 35^3$	$-0.1450^{6.2}$	$-0.0014^{3.1}$
15	$5^0\ 5^2\ 40^0\ 5^2\ 5^0$	$-0.2655^{3.7}$	$-0.1303^{3.7}$
16	$12^0\ 12^1\ 12^0\ 12^1\ 12^0$	$-0.0838^{1.7}$	$-0.1105^{2.1}$
17	$12^0\ 12^1\ 12^2\ 12^3\ 12^4$	$-0.2615^{4.6}$	$-0.1309^{2.6}$
18	$14^3\ 5^5\ 1^0\ 1^2\ 4^5\ 9^2\ 15^3\ 11^4$	$-0.7882^{2.4}$	$-0.2121^{7.0}$
19	$2^1\ 14^0\ 15^1\ 15^4\ 5^1\ 1^5\ 5^0\ 3^1$	$-0.2674^{4.5}$	$-0.2403^{4.4}$
20	$6^0\ 6^2\ 6^0\ 6^2\ 6^0\ 6^2\ 6^0\ 6^2\ 6^0\ 6^2$	$-0.5191^{4.7}$	$-0.4200^{5.8}$

Table 1: Comparisons of BH with CBS and BP for 20 scenes. The superscript is the standard error of the difference in MSE multiplied by 1000.

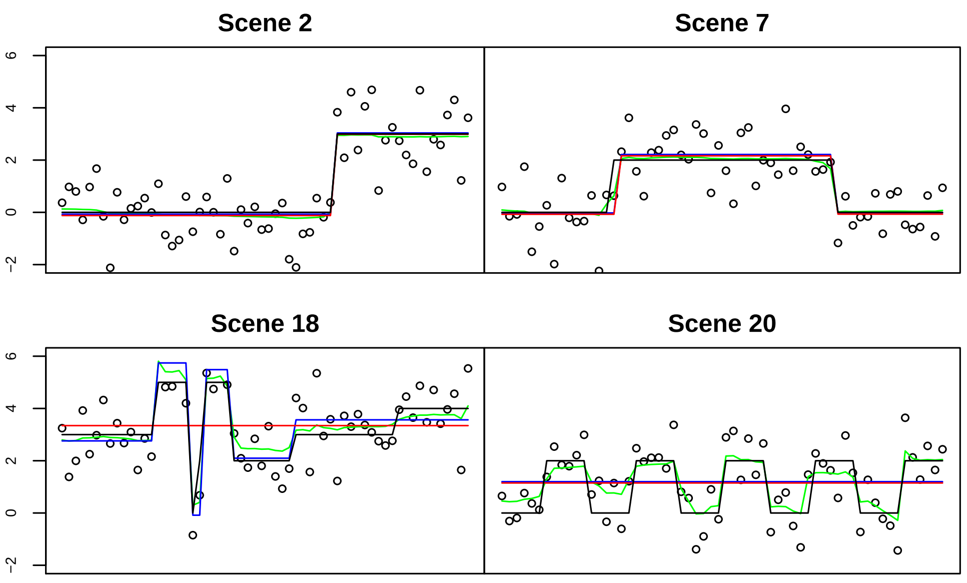

Figure 3: Fitted values for selected scenes. The black lines are the true means, red lines represent CBS, blue lines represent BP, green lines represent BH, and the circles represent one example of simulated data.

seconds for CBS and 3.62 seconds for BP. Our MCMC of the BH procedure is of $O(n^2)$ in speed and $O(n)$ in memory. A future version of **bcp** may be of $O(n)$ in speed. The analysis of microarray data with, for example, 10,000 measurements currently takes approximately 45 minutes. When allowed to look over all partitions, the least squares calculations of the BP dynamic programming algorithm are of order $O(n^3)$ in speed and $O(n^2)$ in memory. Thus, the unconstrained BP procedure is not feasible for large sample sizes. The CBS algorithm is of $O(n)$ in both speed and memory, and takes approximately 10 minutes for a sequence of length 10,000.

BH and CBS have a clear speed and memory advantages over BP which is of essential importance with large data sets. For smaller data sets without memory constraints, however, the BP dynamic programming algorithm is guaranteed to find the optimal least squares solution, while the recursive CBS algorithm is not. Furthermore, a C implementation of the dynamic programming algorithm (designed for the analysis of microarray data, but generally applicable) is available in the **tilingArray** package (Huber and Toedling 2006). For small data sets it is faster than BH, CBS, and the R implementation of BP, but for large data sets, the memory requirements limit its application to problems with frequent change points and blocks of limited length.

Other differences between the procedures are worth noting. Unlike BH and CBS, BP does not assume constant variance. However, a standalone C++ implementation of an extension of BH (allowing for changes in variance) is provided by Loschi and Cruz (2005). A future version of **bcp** may also allow changes in variance, although simultaneously allowing large numbers of change points seems undesirable for most applications. BP provides the flexibility to specify a minimum segment length and/or the maximum number of breaks, but does not allow for single-observation segments (the algorithm requires at least two observations for the least squares calculations). Thus, the procedure will not correctly adapt to the presence of isolated outliers, or true, single-observation blocks. Although CBS and BH do not allow setting the number of breaks or a minimum segment length, BH provides a hyperparameter, `p0`, that may be tuned to encourage shorter or longer segments, at the discretion of the user. As demonstrated in the simulations of Section 3.3, CBS performs best when there are longer segments. Finally, we note that both BP and CBS estimate location(s) of the change point(s); this will appeal to many users. However, the BH Bayesian procedure estimates the probability of a change point at each location, providing a more informative summary reflecting the degree of uncertainty in the change points. We expect this feature to be useful in practice.

References

Bai J, Perron P (2003). "Computation and Analysis of Multiple Structural Change Models." *Journal of Applied Econometrics*, **18**, 1–22.

Barry D, Hartigan JA (1993). "A Bayesian Analysis for Change Point Problems." *Journal of the American Statistical Association*, **35**(3), 309–319.

Erdman C, Emerson JW (2007). **bcp***: A Package for Performing a Bayesian Analysis of Change Point Problems.* R package version 1.8.4, URL `http://CRAN.R-project.org/`.

Finney DJ (1946). "Field Sampling for the Estimation of Wireworm Populations." *Biometrics*, **2**(1), 1–7.

Fridlyand J, Snijders A, Pinkel D, Albertson DG, Jain AN (2004). "Hidden Markov Models Approach to the Analysis of Array CGH." *Journal of Multivariate Analysis*, **90**, 132–153.

Garcia R, Perron P (1996). "An Analysis of the Real Interest Rate Under Regime Shifts." *Review of Econometrics and Statistics*, **78**, 111–125.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "Bioconductor: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**, R80. URL http://genomebiology.com/2004/5/10/R80.

Hamilton JD (1989). "A New Approach to the Economic Analysis of Changes in Regimes: An investigation of the Term Structure of Interest Rates." *Journal Economic Dynamics and Control*, **12**, 385–423.

Huber W, Toedling J (2006). **tilingArray**: *Analysis of High-density Oligonucleotide Tiling Arrays*. R package version 1.10.0, URL http://bioconductor.org/.

Jain AN, Tokuyasu TA, Snijders AM, Segraves R, Albertson DG, Pinkel D (2002). "Fully Automatic Quantification of Microarray Image Data." *Genomics Research*, **12**, 325–332.

Kallioniemi A, Kallioniemi OP, Sudar D (1992). "Comparative Genomic Hybridization for Molecular Cytogenetic Analysis of Solid Tumors." *Science*, **258**(5038), 818–821.

Loschi RH, Cruz FRB (2005). "Extension to the Product Partition Model: Computing the Probability of a Change." *Computational Statistics & Data Analysis*, **48**, 255–268.

Olshen AB, Venkatraman ES (2004). "Circular Binary Segmentation for the Analysis of Array-based DNA Copy Number Data." *Biostatistics*, **5**(4), 557–572.

Plummer M, Best N, Cowles K, Vines K (2007). **coda**: *Output Analysis and Diagnostics for MCMC*. R package version 0.12-1, URL http://CRAN.R-project.org/.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Sen A, Srivastava M (1975). "On Tests for Detecting Change in Mean." *The Annals of Statistics*, **3**(1), 98–108.

Snijders AM, Nowak N, Segraves R, Blackwood S, Brown N, Conroy J, Hamilton G, Hindle AK, Huey B, Kimura K, Law S, Myambo K, Palmer J, Ylstra B, Yue JP, Gray JW, Jain AN, Pinkel D, , Albertson DG (2001). "Assembly of Microarrays for Genome-wide Measurement of DNA Copy Number." *Nature Genetics*, **29**(3), 263–246.

Venkatraman ES, Olshen AB (2006). **DNAcopy**: *A Package for Analyzing DNA Copy Data*. R package version 1.1.0, URL http://www.bioconductor.org/.

Yao YC (1984). "Estimation of a Noisy Discrete-time Step Function: Bayes and Empirical Bayes Approaches." *The Annals of Statistics*, **12**(4), 1434–1447.

Zeileis A, Kleiber C, Krämer W, Hornik K (2003). "Testing and Dating of Structural Changes in Practice." *Computational Statistics & Data Analysis*, **44**, 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/.

Affiliation:

Chandra Erdman, John W. Emerson
Department of Statistics
Yale University
24 Hillhouse Avenue
New Haven, CT 06511
E-mail: Chandra.Erdman@yale.edu, john.emerson@yale.edu
URL: http://www.stat.yale.edu/~cle7/, http://www.stat.yale.edu/~jay/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 23, Issue 3 *Submitted:* 2007-01-02
December 2007 *Accepted:* 2007-11-27