A fast Bayesian change point analysis for the segmentation of microarray data
Chandra Erdman\(^1\)\(^*\) and John W. Emerson\(^1\)
\(^1\)Department of Statistics, Yale University, 24 Hillhouse Avenue, New Haven, CT 06511

ABSTRACT
Motivation: The ability to detect regions of genetic alteration is of great importance in cancer research. These alterations can take the form of large chromosomal gains and losses as well as smaller amplifications and deletions. The detection of such regions allows researchers to identify genes involved in cancer progression, and to fully understand differences between cancer and non-cancer tissue. The Bayesian method proposed by Barry and Hartigan (1993) is well-suited for the analysis of such change point problems. In Erdman and Emerson (2007) we introduce the R package bcp (Bayesian change point), an MCMC implementation of Barry and Hartigan's method. In a simulation study and real data examples, bcp is shown to both accurately detect change points and estimate segment means. Earlier versions of bcp (prior to 2.0) are \(O(n^2)\) in speed and \(O(n)\) in memory (where \(n\) is the number of observations), and run in approximately 45 minutes for a sequence of length 10,000. With the high resolution of newer microarrays, the number of computations in the \(O(n^2)\) algorithm is prohibitively time-intensive.

Results: We present a new implementation of the Bayesian change point method that is \(O(n)\) in both speed and memory; bcp 2.1 runs in approximately 45 seconds on a single processor with a sequence of length 10,000 – a tremendous speed gain. Further speed improvements are possible using parallel computing, supported in bcp via NetWorkSpaces (REvolution Computing). In simulated and real microarray data from the literature, bcp is shown to both accurately detect abberations of varying width and magnitude. The ability to detect regions of genetic alteration is of great importance in cancer research. These alterations can take

1 INTRODUCTION
Numerous publications have stated the importance of accurately detecting regions of genetic alteration. In particular, many methods for the analysis of array comparative genomic hybridization (CGH) data (Kallioniemi et al., 1992) have been developed. In “Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data,” Lai et al. (2005) compare 11 such methods, dividing them into the categories of estimation algorithms and smoothing algorithms. In this paper we present a fast Bayesian alternative – an MCMC implementation of the method proposed in “A Bayesian analysis for change point problems” (Barry and Hartigan, 1993).

Circular Binary Segmentation (CBS) (Olshen and Venkatraman, 2004) is a modification of Binary Segmentation (Sen and Srivastava, 1975). It is an estimation algorithm which uses a likelihood ratio statistic to test the null hypothesis of no change points in a sequence. If the null hypothesis is rejected, the sequence is split and the test is recursively applied to the resulting sub-segments until no additional changes are detected. Jong et al. (2003) introduce a genetic local search algorithm (GA) that searches for the most probable partition of a given number of change points. After smoothing the data, the procedure uses maximum likelihood with a penalty term that increases with the number of change points to select a partition. CGH Segmentation (CGHseg, Picard et al., 2005) also uses a penalized likelihood function to estimate the number of change points in a sequence. Hupe et al. (2004) implement an Adaptive Weights Smoothing procedure (in the R package GLAD) based on the local-likelihood model of Polzehl and Spokoiny (2000). The Quantile Smoothing algorithm (Quantreg, Eilers and Menezes, 2005) is based on the minimization of the sum of absolute errors. The smoothing and estimation algorithm Clustering Along Chromosomes (CLAC, Wang et al., 2005) uses hierarchical clustering to detect change points, and false discovery rate (FDR) for model selection. The estimation algorithm of Fridlyand et al. (2004) is based on a hidden Markov model (HMM). Hsu et al. (2005) propose a nonparametric smoothing algorithm that uses wavelets to “denoise” data. Chromosomal Abberation Region Miner (ChARM, Myers et al., 2004) uses three edge detection filters for smoothing, followed by an EM algorithm for estimation. Lingjardé (2005) presents CGH-Explorer, a program which implements several threshold methods for segmentation, and introduces the smoothing and estimation algorithm ACE (Analysis of Copy Errors). Locally weighted regression and smoothing scatterplots (Lowess), introduced in Cleveland (1979), is also frequently applied to microarray data.

Several issues arise in the comparison of these algorithms in Lai et al. (2005). Some of the estimation algorithms return only the location(s) of the detected abberation(s), and may not provide estimated log-ratios. Some of the smoothing algorithms return estimates of log-ratios without indicating the location(s) of abberation(s). Some algorithms require additional ‘normal vs. normal’ control samples. Some of the algorithms are slow, and some have not been implemented in publicly available software.

The Bayesian approach estimates both the posterior means and the posterior probability of a change for each position in a

\(^*\)to whom correspondence should be addressed
sequence. It is generally applicable and can be used to segment gene expression data, array CGH data, and any other data with contiguous segments of equal mean. The R implementation of the Bayesian approach is available on CRAN as the bcp package for Macintosh, Windows, and Linux. Version 2.0 (or higher) is \( O(n) \) and is often faster than the new \( O(n) \) implementation of CBS (Venkatraman and Olshen, 2007), particularly for long sequences containing at least a few detectable change points. Version 2.1 (or higher) supports parallel MCMC computations via NetWorkSpaces (R package nws) for additional speed improvements.

In section 2 we describe the Bayesian change point model and the MCMC implementation; we also provide a detailed description of the R package bcp. In section 3 we apply the Bayesian approach to the simulated chromosomes used by Lai et al. (2005) to compare the various change point methods, and to the a brain tumor sample from Bredel et al. (2005) (also presented in Lai et al., 2005). Section 4 concludes with a discussion of speed considerations, improvements provided by bcp version 2.0, and directions for future work.

2 METHODS

Implementing Barry and Hartigan’s Bayesian procedure

Barry and Hartigan (1993) propose a Bayesian model for the change point problem – when there is an unknown partition, \( \rho \), of a set into contiguous blocks such that the means are equal within blocks. The model assumes that observations are independent \( N(\mu_i, \sigma^2) \), and that the probability of a change point at a position \( i \) is \( \rho_i \) 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 \( \rho_i + 1 \) is required is that, given the partition and the parameters, observations in different blocks are mutually independent “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 \( \rho_i \) (the mean of the block beginning at position \( i-1 \) and ending at position \( j \)) is chosen as \( N(\mu_0, \sigma^2) \). Note that the variance of this prior changes with the length of the block; specifically, larger deviations from \( \mu_0 \) are expected in shorter blocks. That is, we do not expect to detect a small change in mean if it persists for a short time. However, this choice of prior does allow “weak signals provided that there are sufficient data to estimate them” (Barry and Hartigan, 1993, p. 311). Barry and Hartigan also select independent priors for \( \mu_0, \sigma^2, \) and \( w = \sigma^2/(\sigma^2 + \sigma^2_0) \) (the ratio of signal to error variance). Specifically,

\[
\begin{align*}
\pi(\mu_0) &= 1, \quad -\infty \leq \mu_0 \leq \infty, \\
\pi(\sigma^2) &= 1/\sigma^2, \quad 0 \leq \sigma^2 \leq \infty, \\
\pi(p) &= 1/p_0, \quad 0 \leq p \leq p_0, \\
\pi(w) &= 1/w_0, \quad 0 \leq w \leq w_0, \\
\pi(\rho) &= 1/p_0 \int_0^{p_0} p^{b-1}(1-p)^{n-b} dp,
\end{align*}
\]

where \( p_0 \) and \( w_0 \) are preselected numbers in \([0,1]\) and \( b \) is the number of blocks in the partition.

Although an exact implementation of Barry and Hartigan’s Bayesian procedure is possible, the calculations are \( O(n^2) \); we implement an MCMC approximation that is \( O(n) \). 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 begins with the partition \( \rho = (U_1, U_2, \ldots, U_n) \), where \( n \) is the number of observations and \( U_i = 1 \) 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, \( p_i \), for the conditional probability of a change at the position \( i + 1 \), is obtained from the simplified ratio presented in Barry and Hartigan:

\[
p_i = \frac{P(U_i = 1 \mid X, U_j, j \neq i)}{P(U_i = 0 \mid X, U_j, j \neq i)} = \frac{\int_0^{p_0} p^b (1-p)^{n-b-1} dp}{\int_0^{p_0} p^{b-1} (1-p)^{n-b} dp} \times \frac{\int_0^{w_0} w^b/2 (W_1 + B_1 w)^{(n-1)/2} dw}{\int_0^{w_0} w^{b-1/2} (W_0 + B_0 w)^{(n-1)/2} dw}
\]

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 \( X \) is the data. The tuning parameters \( p_0 \) and \( w_0 \) may take values in \([0,1]\), chosen so that this method “is effective in situations where there aren’t too many changes (\( p_0 \) small), and where the changes that do occur are of a reasonable size (\( w_0 \) small)” (Barry and Hartigan, 1993, p. 312). After each iteration, the posterior means are updated conditionally on the current partition.

The new MCMC implementation benefits from careful updating of the various sums of squares \((W_0, B_0, W_1, \text{ and } B_1)\) in consecutive steps of the algorithm. For example, instead of storing means and variances, the algorithm stores sums and sums of squares within each block of the current partition. When considering the addition (or deletion) of a change point, all quantities within the affected block (or adjacent blocks) are updated directly from these sums and sums of squares.

R package bcp

The bcp package now contains the main `bcp()` function, five methods (`summary()`, `print()`, `plot()`, `fitted()`, and `residuals()`), a new function, `interval.prob()`, for estimating the probability of a change point in an interval, and two datasets. The function `bcp()` performs the analysis, taking six arguments:

- `x`: a numerical vector of data.
- `p0` and `w0`: optional values for Barry and Hartigan’s hyperparameters; 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; following Barry and Hartigan, the default is 500, although longer chains are recommended in practice.
- `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.
- `nwssleigh`: NULL by default. If a sleigh is provided, the `mcmc` steps are run in parallel (see package nws for more information).

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`.

2
• **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 position, 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.

The `interval.plot()` function takes the following arguments:

• **object**: the result of a call to `bcp()`.
• **start**: the starting index of the interval.
• **end**: the ending index of the interval.

It returns the estimated probability of a change point in the interval `[start,end]`. Package `bcp` also contains five methods and two datasets:

• **Methods**:
  
  • The plot method provides two plots summarizing the analysis. The first figure displays the data along with the posterior mean of each position. The second figure shows the proportion of iterations resulting in a change point at each position – the posterior probability of a change.
  
  • The summary and print methods, modeled after 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 with the coda package (Plummer et al., 2007). An example of this conversion is given in the bcp package documentation (Erdman and Emerson (2008)).
  
  • The fitted method returns the vector of posterior means.
  
  • The residuals method returns the data minus the posterior means.

• **Data**:

  • **Coriell**: two array CGH studies of Coriell cell lines, also appears in the DNAcopy package (Venkatraman and Olshen (2007)) that performs CBS, taken originally from Snijders et al. (2001).
  

3 RESULTS

Simulations

We apply `bcp()` to the simulated chromosomes of Lai et al. (2005). Figure 1 shows the results of `bcp()` applied to a particular artificial chromosome with five aberrations of lengths 2, 5, 10, 20, and 40, each with a magnitude of 1, and \( N(0.25^2) \) noise. Figure 1 of Lai et al. (2005, p. 3765) summarizes the results of the 11 aforementioned change point algorithms applied to this simulated chromosome. Of the 11 algorithms, only 3 clearly identify all five aberrations, and two of these three also falsely detect an amplification near position 400. The Bayesian change point algorithm clearly identifies all 5 aberrations and assigns high posterior probability to each, without identifying any false positives.

To examine the tradeoff between detecting true positives and incorrectly identifying false positives, Lai et al. (2005) consider the Receiver Operating Curves (ROC) for the 11 algorithms for aberration widths of 5, 10, 20, and 40, and signal to noise ratios (SNR) of 1, 2, 3, and 4. SNR, as defined by Lai et al. (2005, p. 3765), is “the mean magnitude of the aberration divided by the standard deviation of the superimposed Gaussian noise.” For each combination of aberration width and SNR, they generate 100 artificial chromosomes of length 100, with the aberrations of various widths added in the center. True positive rates (TPR) are calculated as the number of observations inside the region of amplification whose fitted value (posterior mean) is above a threshold, divided by the total number of observations inside the aberration. False positive rates (FPR) are calculated as the number of observations outside the region of amplification whose fitted value above a threshold, divided by the total number of observations outside the aberration. The threshold is varied from the minimum observed value to the maximum observed value.

Figure 2 displays the ROC curves for `bcp()` applied to the simulated chromosomes from Lai et al. (2005) for signal-to-noise ratios of 1 and 2. When the signal-to-noise ratio is 3 or 4 – regardless of aberration width – the points of the ROC curves from `bcp()` are highly concentrated in the extreme upper left corner (indicating a large percentage of true positives, and a small percentage of false positives). As such, we restrict our attention to the more challenging cases with low signal-to-noise ratios. When the SNR drops to a value of 2, there is still a fairly high concentration of points in the upper left corner for aberration widths of 40, 20, and 10, and slightly less for the aberration width of 5. With this low SNR, the TPR for several of the other algorithms takes a fall (see Figure 2 of Lai et al., 2005, p. 3766). It is only when the signal is matched by the noise (SNR = 1) that the Bayesian change point procedure loses its ability to detect the amplifications. However, it is still no worse than the alternative procedures in this difficult case.

Glioblastoma Multiforme example

We apply `bcp()` to a Glioblastoma Multiforme (GBM) sample with a low SNR. GBM is a particularly lethal type of brain tumor, with patients surviving a median of just one year (Lai et al., 2005). Lai et al. (2005) use normalized GBM data from Bredel et al. (2005), and apply the 11 change point algorithms to two GBM samples. The first has a few short amplifications of large magnitude, and the second has one long loss of low magnitude. Because the Bayesian change point method easily detects large amplifications of any length in the simulated chromosomes, we instead present the result of `bcp()` applied to the more challenging GBM sample with a low-magnitude loss (chromosome 13 of GBM31) in Figure 3.

The posterior means in Figure 3 are clearly divided into two regions – the low-amplitude loss from position 0 to about 550 (note that these numbers do not indicate genomic location), and a normal region from position 550 to the end of the chromosome. CGHseq, GLAD, CBS, and GA also detect the loss, dividing the chromosome into these two regions, and detecting up to a dozen “outliers” (see Figure 3 of Lai et al., 2005, p. 3767). The outliers “can either indicate a real focal aberration, some type of polymorphism,
Fig. 1. The results of \texttt{bcp()} applied to the artificial chromosome of Lai et al. (2005, Figure 1, p. 3765). This example contains 5 aberrations of lengths 2, 5, 10, 20, and 40, an amplitude of 1, and $N(0, .25^2)$ noise.

or an experimental artifact” (Lai et al., 2005, p. 3768). ACE and CLAC identify the low-amplitude loss as a series of losses. The estimates provided by the wavelet, quantreg, and lowess smoothing algorithms are generally lower in the first region and higher in the second, but include many local shifts obscuring the view of the global loss. HMM and ChARM do not separate the chromosome into two regions.

4 DISCUSSION

As shown in the simulated and real examples presented here and in Erdman and Emerson (2007), the Bayesian change point analysis is sensitive enough to catch both larger aberrations that are short-lived, and longer aberrations that are of very low magnitude, without significantly increasing the FDR. This is important in the detection of chromosomal aberrations when some may be represented by a single probe or, as in the GBM example (and many tumor samples), the signal is diluted by sample heterogeneity.

In addition to its accuracy, the new implementation of the Bayesian methodology also has a speed advantage. In the simulation study of Erdman and Emerson (2007), we compare the performance of \texttt{bcp()} to that of the original implementation of CBS in DNAcopy version 1.1.0 (Venkatraman and Olshen, 2007). Although the Bayesian approach is shown to consistently outperform CBS in accuracy, the $O(n^2)$ version of \texttt{bcp()} was significantly slower than CBS. Venkatraman and Olshen (2007) present a faster, modified version of CBS that involves a “hybrid p-value” and an early stopping rule. The $O(n)$ implementation of \texttt{bcp()} is now faster than both the original and modified CBS, analyzing a sequence of length 10,000 in approximately 45 seconds, compared to 10 minutes for the original CBS algorithm and several minutes for the modified CBS algorithm (depending on the number of segments in the sequence). Further speed improvements may be realized using \texttt{bcp}'s support for parallel MCMC computations.

The package version 2.0 or higher also includes improved methods for summarization and visualization of the results. The \texttt{plot()} method produces figures identical to 1 and 3; the \texttt{summary()} method produces a detailed statistical summary reflecting the degree of uncertainty in the change points; the \texttt{interval.plot()} function estimates the probability of at least one change point in a specified region of interest. The Bayesian
methodology also provides a natural framework for combining samples. The ability to analyze multiple samples (optionally, in parallel in a multi-core or distributed computing environment) and combine the results will be provided in a future version of the bcp package.

REFERENCES

Fig. 3. The results of bcp() applied to chromosome 13 of the Glioblastoma Multiforme data (Lai et al., 2005; Bredel et al., 2005). The labels on the x-axis indicate relative location, rather than genomic location.