Memorial Sloan-Kettering Cancer Center
Memorial Sloan-Kettering Cancer Center, Dept. of Epidemiology & Biostatistics Working Paper Series

Year 2006 *Paper* 9

A Faster Circular Binary Segmentation Algorithm for the Analysis of Array CGH Data

E S. Venkatraman* Adam Olshen†

*Memorial Sloan-Kettering Cancer Center, venkatre@mskcc.org

†Memorial Sloan-Kettering Cancer Center, Dept. of Epidemiology and Biostatistics, olshena@biostat.ucsf.edu

http://biostats.bepress.com/mskccbiostat/paper9

A Faster Circular Binary Segmentation Algorithm for the Analysis of Array CGH Data

E S. Venkatraman and Adam Olshen

Abstract

Motivation: Array CGH technologies enable the simultaneous measurement of DNA copy number for thousands of sites on a genome. We developed the circular binary segmentation (CBS) algorithm to divide the genome into regions of equal copy number (Olshen {\it et∼al}, 2004). The algorithm tests for change-points using a maximal t-statistic with a permutation reference distribution to obtain the corresponding p-value. The number of computations required for the maximal test statistic is $O(N2),$ where N is the number of markers. This makes the full permutation approach computationally prohibitive for the newer arrays that contain tens of thousands markers and highlights the need for a faster. algorithm.

Results: We present a hybrid approach to obtain the p-value of the test statistic in linear time. We also introduce a rule for stopping early when there is strong evidence for the presence of a change. We show through simulations that the hybrid approach provides a substantial gain in speed with only a negligible loss in accuracy and that the stopping rule further increases speed. We also present the analysis of array CGH data from a breast cancer cell line to show the impact of the new approaches on the analysis of real data.

Availability: An R (R Development Core Team, 2006) version of the CBS algorithm has been implemented in the "DNAcopy" package of the Bioconductor project (Gentleman {\it et∼al}, 2004). The proposed hybrid method for the p-value is available in version 1.2.1 or higher and the stopping rule for declaring a change early is available in version 1.5.1 or higher.

1 Introduction

The DNA copy number at a location in a genome is the number of copies of DNA. The normal copy number is two for the autosomal chromosomes in humans. Chromosomal aberrations in the form of copy number gains or losses are common in cancer and studying them is a way of identifying and validating important cancer genes. For example, Whang-Peng *et al.* [1982] identified deletion in chromosome 3p(14-23) in small cell lung cancer cell lines. Comparative genomic hybridization (CGH) [Kallioniemi *et al.*, 1992, du Manoir *et al.*, 1993] was the first method developed to measure the DNA copy number variation of entire genomes at a 10-20M resolution. Higher throughput techniques based on microarray technology (hence array CGH) have been developed to simultaneously measure DNA copy number at thousands of locations on a genome. Pinkel and Albertson [2005] present a review of the array CGH technologies.

The purpose of these technologies is to study variations in DNA copy number and to identify chromosomal regions that have been gained or lost. We developed the circular binary segmentation (CBS) algorithm [Olshen *et al.*, 2004] to divide the genome into regions of equal DNA copy number. Several alternate algorithms have also been proposed for the analysis of array copy number data. Willenbrock and Fridlyand [2005] in a comparison of algorithms for array CGH data concluded that "*DNAcopy*" (our software implementing the CBS algorithm) "*has the best operational characteristics in terms of its sensitivity and FDR for breakpoint detection.*" Lai *et al.* [2005] conducted a comparison of methods for analyzing array CGH data that included CBS and ten other approaches. They concluded that CBS is one of the two methods that "*appear to perform consistently well.*" Unfortunately, they also found the CBS algorithm to be one of "*the slowest.*" In light of the proven ability of the CBS algorithm to identify the locations of copy number changes it is desirable to improve its speed.

In this manuscript we present two speed enhancements to the original CBS algorithm. The first one is a hybrid approach for the computation of the p-value of the maximal t-statistic using a tail probability approximation for the maxima of a Gaussian random field. The second one is a sequential testing approach for deriving a stopping rule that reduces the number of permutations when there is strong evidence for the existence of a change-point. The enhanced algorithms use the same test statistic as the original to detect change-points but modifies the procedure used to determine whether the change-points are statistically significant. We compare the performance of the new methods to the original permutation

approach both in terms of speed and accuracy on simulated data as well as real data from breast cancer cell lines.

2 Methods

The CBS procedure formulates the analysis of array CGH data as a problem of detecting change-points, where the change-points are the genomic locations of copy number transitions. The algorithm starts with the whole chromosome and segments it recursively by testing for change-points; it stops when none can be found in any of the segments. The test statistic was chosen to enable us to detect a narrow changed segment in the middle of a large segment.

Let $X_1, \ldots, X_m$ be the data corresponding to the m markers for the segment under consideration. The test statistic is the maximal t-statistic given by $T = \max_{1 \le i < j \le m} |T_{ij}|$, where T_{ij} is the two sample t-statistic to compare the mean of the observations with index from $i+1$ to j, to the mean of the rest of the observations. That is

$$T_{ij} = \frac{\bar{Y}_{ij} - \bar{Z}_{ij}}{s_{ij}\{(j-i)^{-1} + (m-j+i)^{-1}\}^{1/2}},$$

where $\bar{Y}_{ij} = (X_{i+1} + \cdots + X_j)/(j-i)$, $\bar{Z}_{ij} = (X_1 + \cdots + X_i + X_{j+1} + \cdots + X_m)/(m-j+i)$, and s_{ij}^2 is the corresponding mean squared error. Note that if we view the segment being tested as indexed by a circle by connecting its two endpoints then the method tests whether there are two complementary arcs that have unequal means. We declare a change to be statistically significant if the p-value is smaller than a threshold level α (typically 0.01) and estimate the locations of the change-points as the i and j (if $j < m$) that maximize the test statistic.

In the original implementation of the CBS algorithm, we compute the p-value using a permutation reference distribution due it being a robust nonparametric method. However, this resulted in the computation time growing quadratically with the number of markers on the array (this was also noticed by Lai *et al.* 2005). This is due to the test-statistic T being the maximum of $m(m-1)/2$ different statistics for every segment considered and m increases with the number of markers on the array. This is computationally burdensome when analyzing high resolution array CGH data because although computing it once is quick it has to be repeated thousands of times to construct the permutation reference distribution. Another source of computational burden is that the procedure computes the entire set of permuted statistics in order to declare that a change-point exists even if part way through the process there is overwhelming evidence for its existence. We will now present a hybrid

method to compute the p-value and derive a stopping rule to declare a change early, both of which will speed up the CBS algorithm substantially.

2.1 Hybrid p-value

The set $\{i, j : 1 \leq i < j \leq m\}$ of all splits considered for the test statistic T can be written for any k ($\leq m/2$) as $A_1 \cup A_2$, where

$$\begin{aligned} A_1 &= \{i, j : j - i \leq k \text{ or } > m - k\}, \\ \text{and } A_2 &= \{i, j : k + 1 \leq j - i \leq m - k\}. \end{aligned}$$

The set A_1 corresponds to all splits in which the minor arc, which is the smaller of the two arcs made by the two point intersection of a circle, contains at most k observations and the set A_2 corresponds to all splits such that both the arcs have more than k observations. Note that $T = \max\{T_1, T_2\}$, where $T_l = \max_{A_l} |T_{ij}|,\ l = 1, 2$. For an observed test statistic value of b, the p-value $P(T > b)$ is bounded below by $P(T_2 > b)$ and, because of the Bonferroni inequality, bounded above by $P(T_1 > b) + P(T_2 > b)$. We compute $P(T_1 > b)$ using a permutation approach since T_1 is the maximum of several correlated t-statistics and an approximation to its distribution is unavailable even for normally distributed Xs. Since T_1 requires only mk statistics to be computed, the computational burden is nearly linear in the number of markers with a suitable choice of k.

We approximate $P(T_2 > b)$ as follows. Heuristically, since T_{ij} is the standardized difference of means, it has a limiting normal distribution under suitable regularity conditions. So for m and k are large enough the distribution of the statistic T_{ij}, under the null hypothesis of no change, is approximately standard normal. Observe that $T_2 = \max\{T_{2+}, T_{2-}\}$ where $T_{2+} = \max_{A_2} T_{ij}$ and $T_{2-} = \max_{A_2} \{-T_{ij}\}$. So under the null hypothesis of no change, the distribution of the statistic T_{2+} is the same as the one if the X_is are independent standard normal. Siegmund [1988] and Yao [1989] independently derived approximations for the tail probabilities of the statistic T_{2+} (this statistic is the same as Z_3 in Yao 1993), which is given by

$$P(T_{2+} > b) \approx \frac{1}{4} b^3 \phi(b) \int_{1/2}^{1-\delta} \frac{\nu^2(b/[mt(1-t)]^{1/2})}{t^2(1-t)^2} dt,$$

where $\delta = k/m$, ϕ is the standard normal density, and ν is defined as

$$\nu(x) = 2x^{-2} \exp\left\{-2 \sum_{l=1}^{\infty} l^{-1} \Phi\left(-\frac{1}{2} x l^{1/2}\right)\right\}.$$

Since by symmetry $P(T_{2+} > b) = P(T_{2-} > b)$ and the probability of both T_{2+} and T_{2-} exceeding b is small, $P(T_2 > b) \approx 2P(T_{2+} > b)$. This approximation is asymptotic, *i.e.* the ratio of the true probability and the approximate formula converges to 1 as $m \to \infty, b \to \infty, b/\sqrt{m}$ a constant greater than 0 and $0 < \delta < 1/2$ fixed. In the next section we show that it is robust and recommend a choice of k as a function of m.

The CBS algorithm is modified in the following manner. Let b be the value of the test statistic T on the observed $X_1, \ldots, X_m$. Since the approximation is asymptotic we compute the full permutation p-value if there are fewer than m_0 markers m. Otherwise, we first compute $P(T_2 > b)$ from the approximation above. If it exceeds α we can declare that there is no change; otherwise we compute $P(T_1 > b)$ and the p-value is the sum of the two and a change is declared if the sum is less than α. The only difference between the hybrid and the permutation approaches is the p-value. The choice of p-value method thus affects whether a change-point is detected but not its estimated location. Since the hybrid p-value is an approximation for an upper bound it can result in fewer change-points detected. We will study empirically, the impact of this on the procedure's ability to detect the true change-points.

2.2 Stopping rule to declare a change

In the CBS algorithm the magnitude of the p-value is relevant only to decide if it exceeds α. The permutation p-value is given by the proportion of times the permuted statistic exceeds the original statistic. Thus the permutations can be stopped and the null of no change accepted as soon as the permuted statistic exceeds the original more than αB times, where B is the number of permutations. However, at least $(1 - \alpha)B$ permuted statistics must be computed to declare that a change exists even if there is overwhelming evidence earlier for it. For example, when $B = 10000$ and $\alpha = 0.01$ the procedure cannot stop and declare that a change is present even if none of the first 1000 permuted statistics exceed the original statistic. We use concepts from sequential testing to derive a stopping rule that declares a change before all the permutations are completed. Such a stopping rule will also benefit the hybrid method since it has a permutation component.

Let $E_1, \ldots, E_B$ be the binary random variables indicating whether the permuted statistic exceeds the original and let $R(i) = E_1 + \cdots + E_i$ be their partial sums. Let r be the smallest integer greater than αB, ($\alpha - P\{T_2 > b\}$ instead of α for the hybrid) that is, r/B is the smallest p-value for which the null hypothesis of no change is not rejected. The stopping

rule is a sequence of integers $b_1 < \cdots < b_r$ such that the permutations are stopped after the b_ith permutation, where i is the smallest j for which $R(b_j) < j$. That is, the permutations are stopped the first time there are fewer than i permuted statistics exceeding the original statistic among the first b_i permutations. The bs are chosen to satisfy

$$P\{R(b_i) < i \text{ for any } 1 \leq i \leq r | R(B) = r\} \leq \eta$$

for a pre-specified (Type I) error rate η. Since there are a large number of boundaries that satisfy the condition we choose the one given by repeated significance testing theory; that is, each b_i is the smallest integer for which $P\{R(b_i) < i | R(B) = r\}$ is less than an $\eta^\star$ that is chosen such that the overall error rate is η. This stopping rule increases the type I error rate of the algorithm by

$$\sum_{r \leq l \leq B} P\{R(b_i) < i \text{ for any } 1 \leq i \leq r | R(B) = l\} \times P(R(B) = l),$$

but the increase is negligible since the summand above decreases very rapidly in l. A heuristic proof of this claim will be given later in this section. As with the hybrid p-value, the stopping rule only affects whether a change-point is detected but not its estimated location. We will now show the derivation of $\eta^\star$ and the corresponding boundary.

Observe that conditioned on $R(B) = r$, the Es are a sequence of $B - r$ zeroes and r ones all of which are equally likely. For notational simplicity, we will omit the conditioning in the following. Note that the set $\{R(j) < i\}$ corresponds to all sequences (Es) with at most $i - 1$ ones among $E_1, \ldots, E_j$; its probability is $\sum_{l=0}^{i-1} \binom{j}{l}\binom{B-j}{r-l} / \binom{B}{r}$. Since $P\{R(j) < i\}$ is decreasing in j and is zero for $j = B$, for any threshold $\eta^\star$, b_i is the smallest j for which the probability gets below $\eta^\star$. The probability of interest $P\{R(b_i) < i \text{ for any } 1 \leq i \leq r\}$, for any boundary $\{b_1, \ldots, b_r\}$, can be obtained as follows. The locations of the r ones are a random sample drawn without replacement from $1, \ldots, B$. Let $L_1 < \cdots < L_r$ be the ordered locations. Since $R(b_i) < i$ if an only if $L_i > b_i$, the probability of interest is $P\{L_i > b_i \text{ for any } 1 \leq i \leq r\}$ which can be written as

$$\sum_{i=1}^{r} P\{L_j \leq b_j \forall\ 0 < j < i, \text{ and } L_i > b_i\}, \tag{1}$$

since $A_1 \cup \ldots \cup A_k = A_1 \cup (A_1^C \cap A_2) \cup \ldots \cup (A_1^c \cap \ldots \cap A_{k-1}^c \cap A_k)$ and the sets on the right are mutually exclusive. This can be calculated exactly using the following recursive equation:

$$\begin{aligned} & P\{L_j \leq b_j \forall\ 0 < j < i, \text{ and } L_i = l_i\} \\ = & \sum_{l_{i-1}=i-1}^{b_{i-1}} P\{L_j \leq b_j \forall\ 0 < j < i-1, \text{ and } L_{i-1} = l_{i-1}\} \\ & \qquad \times P\{L_i = l_i | L_{i-1} = l_{i-1}\} \end{aligned}$$

and $P\{L_i = l_i | L_{i-1} = l_{i-1}\} = \binom{B-l_i}{r-i} / \binom{B-l_{i-1}}{r-i+1}$. Since this can be computationally daunting, we approximate the sum in (1) by

$$\sum_{i=1}^{r} P\{L_j \leq b_j \text{ for } \max(i-d,1) \leq j \leq i-1, \text{ and } L_i > b_i\}, \tag{2}$$

where d is the number of prior points for which the boundary is not crossed. This follows by observing that the sets in the summands of (2) contain those in (1). This is similar in vein to the improved Bonferroni inequalities in Worsley [1982]. Since $P\{R(b_i) < i \text{ for any } 1 \leq i \leq r | R(B) = r\}$ increases as $\eta^\star$ increases, the boundary is obtained using an iterative procedure until the desired error bound η is reached.

We will now give a heuristic proof for the claim that this stopping rule results in a minial increase in the type I error. Under the null hypothesis of no change $R(B)$ has a uniform distribution on $\{0, \ldots, B\}$. Using the Bonferroni inequality we can bound $P\{R(b_i) < i \text{ for any } 1 \leq i \leq r | R(B) = l\}$ by $\sum_{i=1}^{r} P\{R(b_i) < i | R(B) = l\}$. This conditional distribution of $R(b_i)$ is hypergeometric with mean $\mu_l = l \times b_i/B$ and variance $\sigma_l^2 = l(B-l) \times b_i(B-b_i)/B^3$. Recall that i is the $\eta^\star$ quantile of $R(b_i)$ when $l = r$. Thus $(i - \mu_l)/\sigma_l$ is smaller than $\sqrt{l/r}\,(Q - \mu_r)/\sigma_r$, where Q is the $\eta^\star$ quantile of $R(b_i)$. Appealing to normal approximation of the hypergeometric distribution we see that $P\{R(b_i) < i | R(B) = l\}$ decreases rapidly as l increases. Emprically when $l = 2r$, $P\{R(b_i) < i | R(B) = l\}$ is approximately $(\eta^\star)^2$ when $i = 1$ and becomes much smaller as i increases resulting in an excess type I error smaller than $\alpha\eta$.

2.3 Simulation experiments

We conducted simulation experiments to evaluate the performance gain that the improved CBS algorithm provides and the cost in terms of its ability to detect change-points. We first show that the approximation for $P(T_2 > b)$ works well for large m and suitably chosen k and hence could be used to obtain the p-value for the statistic T. We then segment the same simulated data with and without change-points, using the original permutation p-value and the new hybrid by itself and with early stopping. For all these simulations the marker data are generated from standard normal, uniform or beta$(0.5, 1.0)$ distributions. Since the approximation was obtained for the maximum of a Gaussian random field, the normal data is where it is expected to perform the best. The other two were chosen to provide different degrees of non-normality with the uniform having a flat density and the beta a very skewed J-shaped one. The computing times for the procedures give us a measure of the

performance gain and the proportion of times change-points are detected give us the impact on the ability to detect change-points (false positive or missed ones). All computations were done on a 3.2GHz Intel Xeon PC running the Debian Linux operating system with gcc and g77 compilers and R statistical software.

3 Results

3.1 Evaluating the approximation

We generated $100,000$ data sets with m markers, where m is one of $100, 200, 500$ or 1000 from each of the three distributions. We calculated the statistic T_2 for each of the data sets with $k = 25$. This choice of k was used so that the differences in means in T_{ij} are based on sufficiently large number of observations for a normal approximation to be reasonable. We computed the tail probability $P(T_2 > b)$ empirically as well as by using the approximation for a range of bs for each m. The results are shown in Figure 1. Observe that the approximation substantially underestimates the tail probability when m is only 100. For larger values of m the approximation is close for all three marker distributions. Hence we recommend a minimum of 200 markers (the recommended value of m_0 in Section 2.1) when using the hybrid approach. Since the full permutation approach for this number of markers can be accomplished with modest computing effort, this requirement is not a serious burden.

The tail probability approximation is an asymptotic result with k/m a constant. Thus one concern is the use of a fixed k for all values of m. In order to assess this we chose

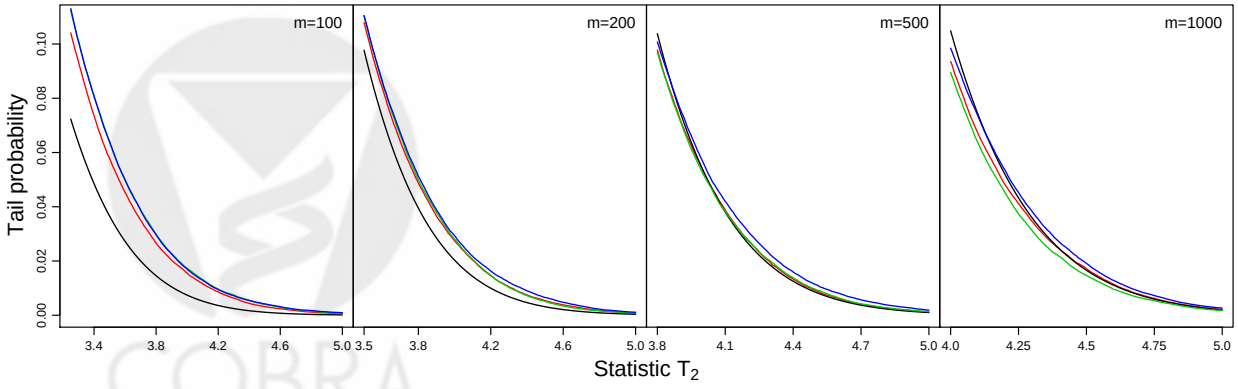

Figure 1: The empirical tail probability distribution of the statistic T_2 when the data are normal (red), uniform (green) or beta$(0.5, 1.0)$ (blue) for $m = 100, 200, 500,$ or 1000 and $k = 25$. The black line is the approximation.

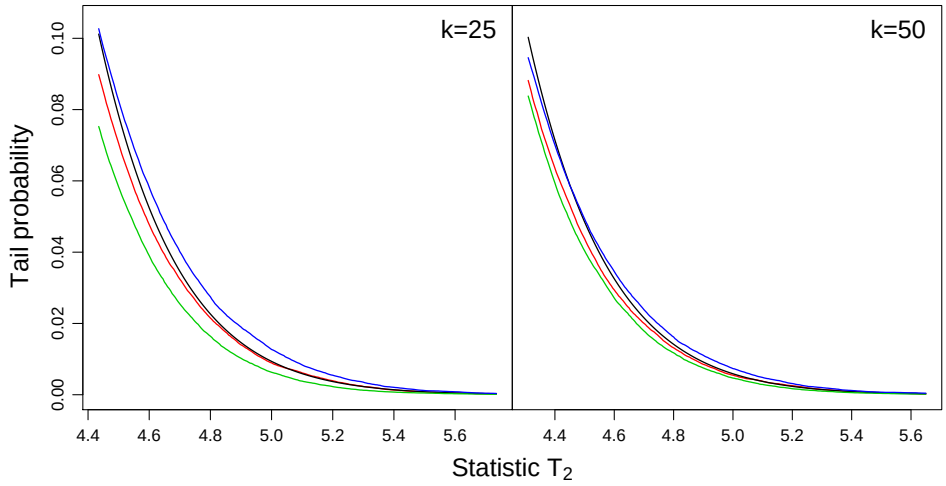

Figure 2: The empirical distribution of the statistic T_2 when the data are normal (red), uniform (green) or beta$(0.5, 1.0)$ (blue) for 5000 markers and $k = 25$ or 50.

m to be 5000 and evaluated the tail probability empirically when k is 25 or 50. This is shown in Figure 2, along with the approximation. Observe that there is a larger difference between the empirical values and the approximation when k is 25 while the degree to which the approximation agrees with the empirical values when k is 50 is similar to that found in the $m = 1000$ panel of Figure 1. These simulations show that with a properly chosen k the approximation for $P(T_2 > b)$ works well for larger m and could be used to obtain the p-value for the segmentation procedure as described in the previous section. This also suggests that it would be prudent to choose a larger k when m increases. We recommend that k be set at 25 for m smaller than 1000 and be increased in increments of 5 as the number of markers doubles.

3.2 Performance of the segmentation procedure

In the previous section we showed that the approximation to $P(T_2 > b)$ works well. We will now evaluate the operating characteristics of the CBS algorithm when the p-value is approximated by $P(T_1 > b) + P(T_2 > b)$ as well as when the stopping rule to declare a change early is included. Specifically, we wish to assess the reduction in computing time achieved by these modifications and their cost in terms of false positive or missed change-points. We compare the CBS algorithms with the hybrid p-value alone, as well as when combined with the stopping rule to declare a change early, to the original that uses the full permutation p-value.

Table 1: The results of segmentation when there is no change in the data. The *size* column gives the percent of data sets when change-points were detected (false positives) and the *time* column gives the user cpu time in minutes to segment the 5000 data sets. The top and bottom halves correspond to $\alpha = 0.01$ and 0.05 respectively. The N, U and B in the first column indicate the marker distribution. Permutation used the original full permutation p-value, Hybrid used the approximation and Hybrid + ES used the approximation and early stopping.

		$m = 250$		$m = 500$		$m = 1000$	
		size	time*	size	time*	size	time*
N	Permutation	1.04	36.6	0.92	124.6	1.08	500.1
	Hybrid	1.08	11.2	0.88	18.7	1.00	35.3
	Hybrid + ES	1.08	10.2	0.84	17.2	1.08	32.3
U	Permutation	1.12	36.6	1.00	132.5	1.22	492.1
	Hybrid	1.16	11.4	0.90	19.1	1.00	34.5
	Hybrid + ES	1.18	10.3	0.94	17.2	1.08	31.8
B	Permutation	0.86	35.4	0.86	128.6	1.14	498.0
	Hybrid	1.04	11.1	0.92	19.4	1.08	39.5
	Hybrid + ES	1.04	10.5	0.92	17.8	1.06	36.1
N	Permutation	5.44	121.1	4.56	441.7	4.72	1749.7
	Hybrid	5.22	22.4	4.20	36.5	4.44	68.6
	Hybrid + ES	5.20	16.9	4.24	27.6	4.46	50.9
U	Permutation	5.24	120.7	5.08	451.2	4.60	1710.4
	Hybrid	4.82	19.9	4.44	32.8	3.64	49.8
	Hybrid + ES	4.92	14.9	4.50	23.4	3.64	39.5
B	Permutation	5.04	119.5	5.12	451.6	4.96	1765.7
	Hybrid	4.94	21.8	4.80	35.6	4.64	68.7
	Hybrid + ES	5.02	16.7	4.80	27.2	4.54	49.3

* elapsed times on a 3.2GHz Pentium 4 computer.

We simulated the case of no change-points by generating 5000 data sets each for each of $250, 500$ or 1000 markers and for each of the three distributions. The segmentations were performed using a p-value threshold (α) of 0.01 or 0.05. Table 1 shows the percent of times the procedure segments the data (size) and the CPU time used. The size values are very similar for the three algorithms and are consistent with the nominal α level. The size of the

Table 2: Segmentation results when there is a change. The *power* column gives the percent of data sets when change-points were detected (100 - power is the percent of missed change-points). The description of the other elements of the table is the same as the one for Table 1. The times are the total time to segment the 1000 data sets of each type.

		$m = 250$		$m = 500$		$m = 1000$	
		power	time*	power	time*	power	time*
N	Permutation	79.8	102.9	80.1	380.6	78.2	1509.0
	Hybrid	80.3	26.0	80.0	47.0	78.0	91.5
	Hybrid + ES	80.4	8.6	79.9	13.0	78.3	26.4
U	Permutation	80.0	102.8	78.6	379.9	75.7	1485.3
	Hybrid	80.5	25.8	78.4	45.6	74.8	86.3
	Hybrid + ES	80.6	7.9	78.3	12.0	74.7	22.3
B	Permutation	76.7	99.5	76.8	378.6	72.7	1450.9
	Hybrid	77.1	24.8	76.8	46.3	73.0	86.6
	Hybrid + ES	77.0	8.3	77.0	14.2	72.7	26.7
N	Permutation	92.1	121.2	87.8	441.8	89.1	1830.8
	Hybrid	91.7	36.8	87.7	54.7	88.4	110.2
	Hybrid + ES	91.6	13.6	87.5	13.6	88.7	28.5
U	Permutation	91.5	119.7	89.1	445.3	88.4	1824.0
	Hybrid	91.1	36.0	88.0	54.2	86.8	107.6
	Hybrid + ES	91.2	13.1	88.1	12.8	86.8	25.6
B	Permutation	90.1	119.5	90.3	452.3	87.3	1823.9
	Hybrid	89.8	35.9	90.0	56.3	86.9	108.5
	Hybrid + ES	90.0	13.6	90.0	14.3	87.2	28.2

* elapsed times in minutes on a 3.2GHz Pentium 4 computer.

new algorithms are lower that of the original, which is consistent with the hybrid p-value being an approximate upper bound. Also, the addition of the stopping rule results in only a negligible increase in the significance level of the test as the achieved size barely increases. However the newer algorithms show an enormous gain in computing speed. For example, the normal data with 1000 markers and $\alpha = 0.05$ took 29 hours using the original algorithm but only a bit over an hour with the hybrid algorithms. Since the stopping rule only enables us to declare a change early, we do not expect it to affect the time taken when there is no change and the results confirm it. Notice also that the computing times for the original

algorithm quadruples when the number of markers doubles, whereas it grows linearly for the hybrid algorithms.

For the alternative of there being change-points, we generated 1000 data sets each and added μ to 10 contiguous markers in the middle of each set, where μ is chosen (empirically) to give approximately 80% power of detecting the change when $\alpha = 0.01$. These results are in Table 2, with power being the percent of times the change was detected. The power of all three procedures are comparable with the original only minimally more powerful than the hybrid algorithms. Thus the fact that the size of the hybrid procedure could be moderately conservative has minimal impact on the power of the procedure. As seen in the no change case the hybrid algorithms show enormous speed gains over the original. The addition of the stopping rule makes the procedure 3 to 4 times faster than the hybrid alone.

A final set of simulations were done to represent the case of multiple change-points in a choromosome. We reproduce a subset of the cases considered in the second set of simulations in Olshen *et al.* [2004]. There are 497 markers in the chromosome with 6 change-points and the mean log-ratios (μ) for the markers given by:

Marker	1-	138-	225-	242-	299-	308-	332-
i	137	225	241	298	307	331	497
μ_i	-0.18	0.08	1.07	-0.53	0.16	-0.69	-0.16

One thousand observed logratio data sets were generated as $c * \mu + Z$ where Z is standard normal and the scale factor c is 10 or 5 to represent low and high noise scenarios. The data were segmented using the permutation, the hybrid and the hybrid with early stopping procedures using $\alpha = 0.01$ and 0.05. The number of change-points detected by the three procedures are reported in Table 3 and show that the three are nearly identical. These numbers are consistent with the results in Olshen *et al.* [2004]. The number of data sets for which all three procedures identified the same set of change-points are

α, c	(0.01, 10)	(0.01, 5)	(0.05, 10)	(0.05, 5)
identical	985	975	977	963

The differences between the three procedures are minimal and could be partly explained by the differences in the stream of random numbers in their permutation component.

These simulation results clearly demonstrate that the new CBS algorithm with the hybrid p-value and stopping rule for delaring change early provide speed gains that make CBS a more practical approach for analyzing high resolutions arrays.

Table 3: The number of change-points detected by the three methods. The times are the total time to segment the 1000 data sets of each type.

c	α	#	6	7	8	9	10	11+	time*
10	0.01	P	918	45	36	1	0	0	667.5
		H	914	46	39	1	0	0	123.5
		H+ES	915	48	36	1	0	0	20.5
10	0.05	P	712	67	167	24	23	7	684.2
		H	719	66	161	24	23	7	139.7
		H+ES	709	69	167	24	24	7	23.3
5	0.01	P	859	103	35	3	0	0	669.2
		H	859	103	34	3	1	0	125.9
		H+ES	859	99	38	4	0	0	31.1
5	0.05	P	572	190	168	41	21	8	685.4
		H	574	184	169	42	24	7	142.1
		H+ES	571	190	170	40	21	8	33.7

* elapsed times in minutes on a 3.2GHz Pentium 4 computer.

4 Example

In this section we present two analyses of data from cell lines in order to evaluate how the improved algorithms perform on real data. The first is the analysis of the data from the ROMA [Lucito *et al.*, 2003] array example shown in Figure 4 of Olshen *et al.* [2004]. There are 9820 markers in this array with the maximum of 824 on chromosome 2. The data were segmented with all 3 procedure at an α level of 0.01. We used the hybrid p-value when the segment being tested had more than 200 markers and used $k = 25$. All 3 algorithms found the same 48 segments in the genome as can be seen in Figure 3. The original CBS algorithm segmented the data in 347 seconds, which was reduced to 44 seconds with the hybrid p-value and further reduced to 13 seconds with the inclusion of the stopping rule to declare a change early.

The second example is the analysis of data from 3 breast cancer cell lines (MCF7, SKBR3, and ZR75) evaluated using the Affymetrix 100k SNP platform. These data are available on the SNPscan website (`http://pevsnerlab.kennedykrieger.org/snpscan.htm`; Ting *et al.* [2006]). These 100k SNP array data have a total of 115571 markers over 23 chromosomes (none from Y) with chromosome 2 having the maximum number (10352). The data were

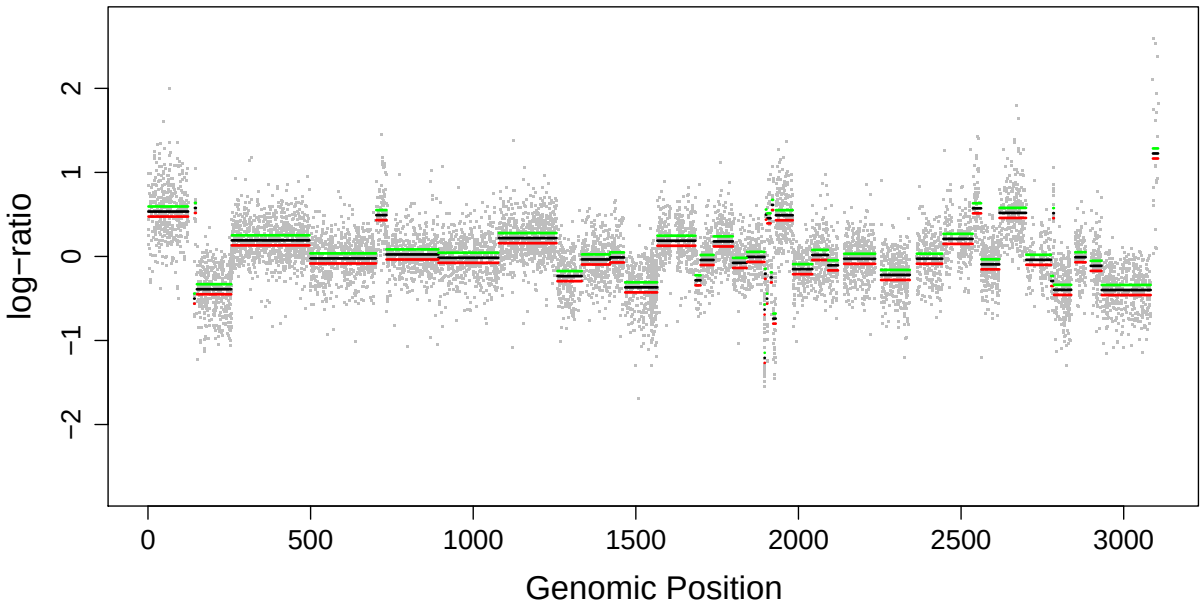

Figure 3: The ROMA array on a breast cancer cell line shown in Olshen *et al.* [2004]. The segments by the original full permutation p-value (black), the new hybrid p-value (red) and the hybrid p-value along with the early stopping to declare change (green).

segmented using all three procedures. The numbers of change-points detected, the number identical and the total computing time per procedure are in the following table.

	Cell line			Time in
	MCF7	SKBR3	ZR75	minutes
P	220	242	270	7085.8
H	217	242	271	100.2
H+ES	216	243	269	25.0
Identical	213	242	263	

Detailed breakdown of the number of change-points by cell line and chromosome are provided in Table 4. This example demonstrates the speed gain (total computing time for the 3 cell lines is nearly 5 days for the original versus 100 minutes for hybrid alone and 25 minutes for the hybrid with early stopping) the new procedures accomplish without sacrificing efficacy.

Table 4: The number of change-points detected by the permutation (P), the hybrid (H) and the hybrid with early stopping (H+ES) procedures. The numbers that are identical across all three methods are also given.

	Cell line MCF7				**Cell line SKBR3**				**Cell line ZR75**			
Chromosome	P	H	H+ES	Identical	P	H	H+ES	Identical	P	H	H+ES	Identical
1	38	38	37	37	8	8	8	8	8	8	8	8
2	12	12	12	12	3	3	3	3	3	3	3	3
3	19	19	19	19	34	34	34	34	34	34	34	34
4	3	3	3	3	11	11	11	11	11	11	11	11
5	4	4	4	4	19	19	19	19	19	19	19	19
6	6	6	6	6	11	11	11	11	11	11	11	11
7	14	15	15	14	13	13	13	13	13	13	13	13
8	20	19	19	19	64	64	65	64	64	64	65	64
9	8	8	8	8	1	1	1	1	1	1	1	1
10	6	4	6	4	14	14	14	14	14	14	14	14
11	12	12	12	12	2	2	2	2	2	2	2	2
12	9	9	9	9	6	6	6	6	6	6	6	6
13	12	11	9	9	4	4	4	4	4	4	4	4
14	3	3	3	3	10	10	10	10	10	10	10	10
15	9	9	9	9	3	3	3	3	3	3	3	3
16	2	2	2	2	2	2	2	2	2	2	2	2
17	12	12	12	12	16	16	16	16	16	16	16	16
18	3	3	3	3	6	6	6	6	6	6	6	6
19	1	1	1	1	4	4	4	4	4	4	4	4
20	18	18	18	18	9	9	9	9	9	9	9	9
21	3	3	3	3	0	0	0	0	0	0	0	0
22	3	3	3	3	1	1	1	1	1	1	1	1
X	3	3	3	3	1	1	1	1	1	1	1	1

5 Discussion

We developed the CBS algorithm as a robust non-parametric method for segmenting array CGH data in order to facilitate the identification of chromosomal regions of gain or loss. This method has been successfully applied in several studies to characterize the copy number variation in different types of cancer [Aguirre *et al.*, 2004, Brennan *et al.*, 2004, Chen *et al.*, 2006, Zhao *et al.*, 2005]. In their comparative study Lai *et al.* [2005] found that the CBS algorithm performed consistently well but was slow. In this manuscript we presented the hybrid approach to compute the p-value for the maximal statistic, and added a stopping rule to declare a change early, in order to increase speed. The simulation study we conducted shows that the improved methods perform as well as the full permutation approach and results in the desired speed gains. The similarity of the results among the methods was demonstrated on real data from breast cancer cell lines. We conclude that the CBS algorithm is a natural choice for the analysis of high resolution array CGH data.

Acknowledgement

We thank Mike Wigler for the breast cancer cell line data and Glenn Heller for helpful discussions.

References

Aguirre,A.J. *et al.* (2004) High-resolution characterization of the pancreatic adenocarcinoma genome. *Proc. Natl. Acad. Sci. USA.*, **101**, 9067-9072.

Brennan,C. *et al.* (2004) High-resolution global profiling of genomic alterations with long oligonucleotide microarray. *Cancer Research*, **64**, 4744-4748.

Chen,W. *et al* (2006). Array comparative genomic hybridization reveals genomic copy number changes associated with outcome in diffuse large B-cell lymphomas. *Blood*, **107**, 2477-2485.

du Manoir,S. *et al* (1993) Detection of complete and partial chromosome gains and losses by comparative genomic in situ hybridization. *Human Genetics*, **90**, 590-610.

Gentleman,R.C. *et al.* (2004) Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, **5**, R80.

Kallioniemi,A. *et al.* (1992) Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, **258**, 818-821.

Lai,W.R. *et al.* (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, **21**, 3763-3770.

Lucito,R. *et al.* (2003). Representational oligonucleotide microarray analysis: a highresolution method to detect genome copy number variation. *Genome Research*, **13**, 2291-2305.

Olshen,A.B. *et al.* (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557-572.

Pinkel,D. and Albertson,D. (2005) Array comparative genomic hybridization and its application in cancer, *Nature Genetics*, **37**, S11-S17, Suppl. S.

R Development Core Team. (2006) *R: A Language and Environment for Statistical Computing.* Vienna, Austria. R Foundation for Statistical Computing.

Siegmund,D.O. (1988) Approximate tail probabilities for the maxima of some random fields, *Annals of Probability*, **16**, 487-501.

Ting,J.C. *et al.* (2006) Analysis and visualization of chromosomal abnormalities in SNP data with SNPscan. *BMC Bioinformatics*, **7**:25

Whang-Peng,J. *et al.*, (1982) Specific chromosome defect associated with human small-cell lung cancer; deletion 3p(14-23). *Science*, **215**(4529), 181-182.

Willenbrock,H. and Fridlyand,J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, **21**, 4084-4091.

Worsley,K.J. (1982) An improved Bonferroni inequality and applications. *Biometrika*, **69**, 297-302.

Yao,Q. (1989) Large deviations for bounary crossing probabilities of some random fields. *J. Math, Res. Exposition*, **9**, 181-192.

Yao,Q. (1993) Tests for change-points with epidemic alternatives, *Biometrika*, **80**, 179-191.

Zhao,X.J. *et al.* (2005) Homozygous deletions and chromosome amplifications in human lung carcinomas revealed by single nucleotide polymorphism array analysis. *Cancer Research*, **65**, 5561-5570.