*doi:10.1093/nar/gkr137*

# An all-statistics, high-speed algorithm for the analysis of copy number variation in genomes

**Chih-Hao Chen[1,2], Hsing-Chung Lee[1,3], Qingdong Ling[1,2], Hsiao-Rong Chen[1], Yi-An Ko[1], Tsong-Shan Tsou[1,2,4], Sun-Chong Wang[1], Li-Ching Wu[1] and H. C. Lee[1,2,5,6,*]**

[1]Graduate Institute of Systems Biology and Bioinformatics, National Central University, Chungli, Taiwan 32001, [2]Cathay Medical Research Institute, [3]Department of Surgery, Cathay General Hospital, Taipei, Taiwan 10630, [4]Graduate Institute of Statistics, National Central University, [5]Department of Physics, National Central University, Chungli, Taiwan 32001 and [6]National Center for Theoretical Sciences, Shinchu, Taiwan 30043

Received September 27, 2010; Revised February 5, 2011; Accepted February 25, 2011

## ABSTRACT

**Detection of copy number variation (CNV) in DNA has recently become an important method for understanding the pathogenesis of cancer. While existing algorithms for extracting CNV from microarray data have worked reasonably well, the trend towards ever larger sample sizes and higher resolution microarrays has vastly increased the challenges they face. Here, we present Segmentation analysis of DNA (SAD), a clustering algorithm constructed with a strategy in which all operational decisions are based on simple and rigorous applications of statistical principles, measurement theory and precise mathematical relations. Compared with existing packages, SAD is simpler in formulation, more user friendly, much faster and less thirsty for memory, offers higher accuracy and supplies quantitative statistics for its predictions. Unique among such algorithms, SAD's running time scales linearly with array size; on a typical modern notebook, it completes high-quality CNV analyses for a 250 thousand-probe array in ~1 s and a 1.8 million-probe array in ~8 s.**

## INTRODUCTION

Amplification or deletion of chromosomal segments can lead to abnormal mRNA transcript levels and results in malfunctioning of cellular processes. Locating such chromosomal aberrations in comparative genomic DNA samples, or copy number variation (CNV) (1–4), is an important step in understanding the pathogenesis of many diseases, especially cancer. Array comparative genomic hybridization (CGH) is a high-throughput technique developed for measuring such changes (5–7). CGH arrays using Bacterial Artificial Chromosome (BAC) clones have resolutions of the order of 1Mb (6). Those using cDNA and oligonucleotide as probes (1,8) are less robust than BACs for large segments, but offer much higher resolutions (in the order of 50–100kb). In particular, oligonucleotide arrays allow design flexibility and greater coverage and provide good sensitivity (8). Tiling on custom arrays is also available now for even finer resolution of specific regions and allow the detection of micro-amplifications and deletions (9,10). The drastic improvement in resolution has led to a corresponding increase in the number of probes on an array; modern high-resolution arrays now easily exceed one million probes. Such arrays exact a severe requirement on the speed and accuracy of algorithms used to analyze them and have vastly reduced the usefulness of existing algorithms that are $\mathcal{O}(N^2)$—$N$ is array size—in computation time or memory requirement. Here, we propose a novel algorithm, segmentation analysis of DNA (SAD), for studying CNV in high-resolution arrays.

For a probe, the log2-ratio of intensities from a pair of microarrays is termed a datum. Based on our observation that datum errors tend to be normally distributed, we designed SAD with three features, respectively involving the use of: (i) the Gaussian distribution function (Gaussian) as a probability density function (PDF) for evaluating the true value of a measured datum; (ii) a clustering procedure based on a technique we call pair-wise Gaussian merging (PGM); (iii) $z$-statistic for making clustering decisions. Details are given in Methods. The operational principles of PGM are schematically illustrated

[*]To whom correspondence should be addressed. Tel: +886 3 4227151 (Ext. 36104); Fax: +886 3 4273822; Email: hclee12345@gmail.com
Present addresses:
Hsiao-Rong Chen, Graduate Program in Molecular Medicine, Boston University School of Medicine, Boston, MA 02118, USA.
Yi-An Ko, Graduate Division of Biomedical Sciences, Albert Einstein College of Medicine, Bronx, NY 10461, USA.

in Figure 1. In this case, the original 10 datums are predicted by SAD to have an underlying structure of two segments. SAD has one essential parameter, the threshold $z$-value $z_0$, and an optional one, the sampling size $N_s$. $z_0$ defines a significance level $p_0$ for making clustering decisions and for calling CNVs. $N_s$ is used for speeding up SAD.

We show in the following sections that, compared with algorithms found in the literature, SAD has a simpler but more rigorous formulation, is easier to understand and simpler to use, provides clearer statistical interpretation for its results, requires less memory, offers better accuracy and is vastly faster in computation speed.

## MATERIALS AND METHODS

### Normal distribution of error

Data not having any CNV are best for demonstrating normal distribution of error. For this reason we contrasted pairs of replicate arrays among each of the four triplicate array sets, NA15510_Nsp, NA15510_Sty, NA10851_Nsp and NA10851_Sty (henceforth the Redon data set), that were produced on the Affymetrix 500K EA platform in a CNV study (3). Because each set has three contrasted pairs, the sets give a total of 12 error distributions. In Figure 2, the error distributions, after standardization and normalization, are compared to standard normal distributions in terms of the Kolmogorov–Smirnov statistic (KS). The small KS values confirm that Gaussian is an excellent approximation to the error distributions.

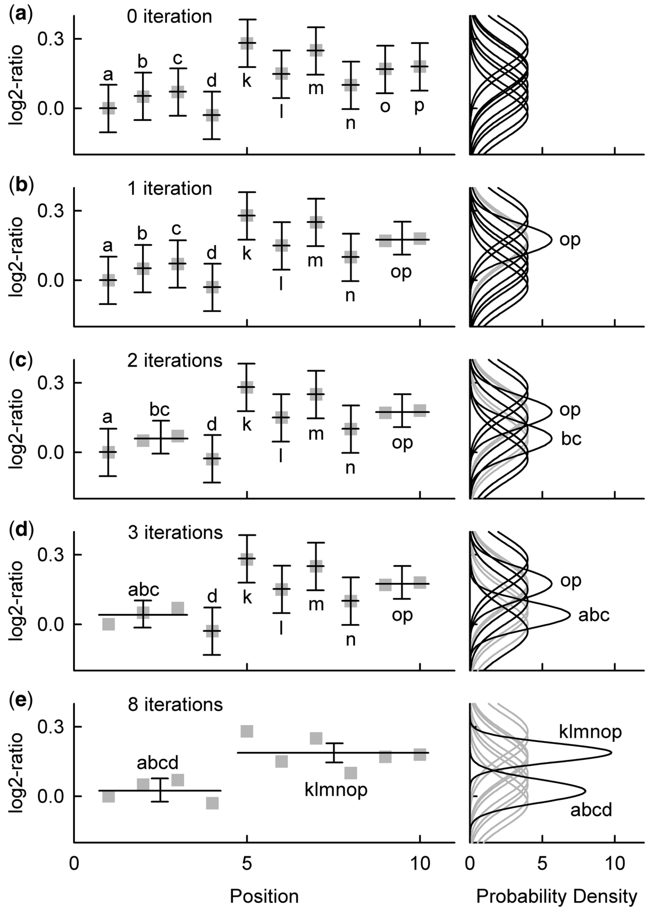

Figure 1. Schematic illustration of PGM applied to genome segmentation. Frames on the left, with the $x$-axis indicating relative probe position on the genome, display datums, as solid grey squares and clusters, as black crosses with errorbars; frames on the right display associated Gaussians. (a) Each datum is treated as a Gaussian with same variance. (b) Datums 'o' and 'p', the nearest neighbouring pair, are merged in the first iteration. (c) and (d) Second and third iterations, respectively. (e) Merging stops after eight iterations when the remaining pair of clusters are considered resolvable.

We examined error properties in more detail using the Affymetrix 500K copy number sample data set (http://www.affymetrix.com). Figure 3a shows the log2-ratio profile of chromosome 2 from the (CRL-5868D, CRL-5957D) STY pair and our selection of two ~8000-datum sections of obviously distinct means. Figure 3b compares the log2-ratio distributions of the two sections with their respective Gaussian approximations, $G(y;0.35,(0.22)^2)$ and $G(y;-0.13,(0.23)^2)$, which have different means but similar variances. These two sections and an artificial 8000-datum section of randomly generated $G(y;0,(0.22)^2)$ noise were used to study the sample-size and spatial dependence of error. Each section is partitioned into subsections of width $4^i$, $i = 3$ to 8, plus a discarded remainder. The error of each subsection is measured using Equation (4). Each section at each $i$ thus has an error distribution whose mean and standard deviation are plotted in Figure 3c. The two sections are shown to have spatial as well as statistical

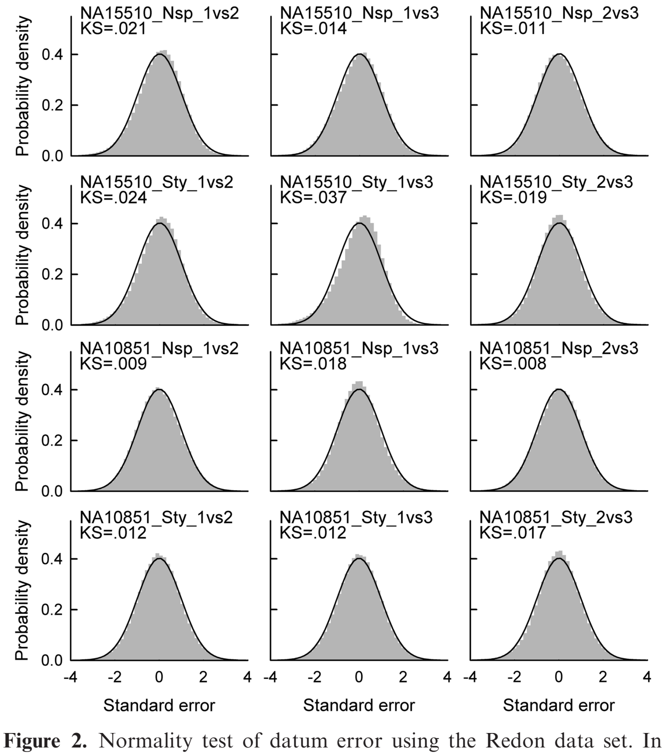

Figure 2. Normality test of datum error using the Redon data set. In terms of KS, the normalized standard error distributions, shown as grey histograms, are compared to standard normal distributions, shown as black lines.

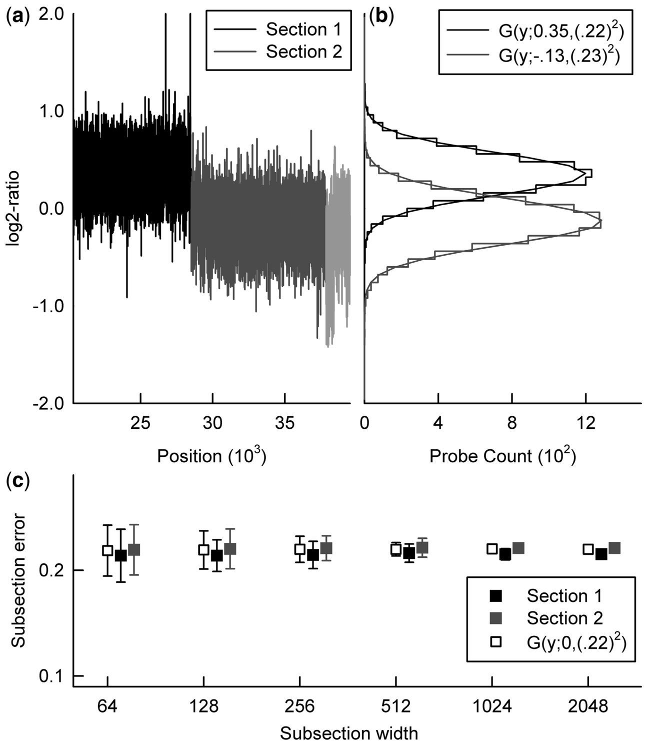

**Figure 3.** Sample-size and spatial independence of variation. Data are from the Affymetrix 500K copy number sample data set. (**a**) The 2 sections and a remainder of chromosome 2 from the (CRL-5868D,CRL-5957D) STY pair. (**b**) log2-ratio distributions of sections 1 and 2 compared with their Gaussian approximations. (**c**) Subsection error distributions computed from subsections of the two sections and, for comparison, an artificial 8000-datum section generated with Gaussian noise.

properties similar to that of the artificial data. In particular, this implies that, for the array data, statistical errors (excluding breakpoints) are more or less uniformly distributed.

## Pair-wise Gaussian merging

Given a measured value $v$, the conditional probability for its true value being $y$ is $Pr(y|v) = Pr(y \cap v)/Pr(v)$. Similarly, given a set of independently measured values $\Omega = \{v_i | i = 1, \ldots, w\}$, we have $Pr(y|\Omega) = Pr(y \cap \Omega)/Pr(\Omega)$ and, from the independence of events, $Pr(y \cap \Omega) = \prod_{i=1}^{w} Pr(y \cap v_i)$, $Pr(\Omega) = \prod_{i=1}^{w} Pr(v_i)$. Therefore, $Pr(y|\Omega) = \prod_{i=1}^{w} Pr(y|v_i)$. In case of continuous variables, the probability that the true value lies in the interval $y$ to $y+dy$ is $Pr(y;dy|v) = dyD(y|\Omega)$, with $D(y|\Omega) \propto \prod_{i=1}^{w} D(y|v_i)$, where the $D$'s are PDFs. Given that errors are normally distributed with initial variance $\tilde{\sigma}^2$, we approximate $D(y|v_i)$ by a Gaussian $G(y;v_i,\tilde{\sigma}^2) = (\tilde{\sigma}\sqrt{2\pi})^{-1}\exp(-(y-v_i)^2/2\tilde{\sigma}^2)$. Repeatedly using the relation that a product of two Gaussians is another Gaussian we have

$$G(y;\mu_1,\sigma_1^2)G(y;\mu_2,\sigma_2^2) \propto G(y;\mu,\sigma^2);$$

$$\frac{\mu}{\sigma^2} = \frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}; \quad \frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}, \tag{1}$$

$$d(y|\Omega) = G(y;\mu,\sigma^2); \quad \mu = \sum_{i=1}^{w} v_i/w; \quad \sigma^2 = \tilde{\sigma}^2/w. \tag{2}$$

We call this method of merging Gaussians to obtain a PDF from a set of measurements Gaussian merging (GM). The formulations of both $\mu$ and $\sigma$ are intuitively understood: $\mu$ is the mean of the measured values and $\sigma^2$ is inversely proportional to sample size, as expected.

To allow the possibility that $\Omega$ comprises multiple subsets each the manifest of a different true value, we conduct a two-sample $z$-test (for independent samples with equal variances), before merging two Gaussians using a $z$-value, here called the resolvability,

$$z_r(G_1,G_2) \equiv \frac{\mu_1 - \mu_2}{\tilde{\sigma}}\left(\frac{1}{w_1} + \frac{1}{w_2}\right)^{-1/2}, \tag{3}$$

where $G_k \equiv G(y;\mu_k,\sigma_k^2)$, $k = 1$ and 2. That $z_r$ follows a standard normal distribution is shown in Supplementary Data. The corresponding $P$-value of $z_r$ tests the null hypothesis that $G_1$ and $G_2$ have the same true value. Given threshold resolvability $z_0$, we say $G_1$ and $G_2$ are resolvable if $|z_r(G_1,G_2)| \geq z_0$, in which case the two Gaussians are kept separate, and are unresolvable and merged otherwise. The following four-step procedure, which we call PGM, partitions $\Omega$ into resolvable subsets: (i) Estimate the variance of each datum. (ii) Select $z_0$. (iii) Identify the unresolvable pair of Gaussians with the smallest $z_r$ and use GM to merge the pair. (iv) Iterate step (iii) until all remaining pairs are resolvable. PGM is a type of agglomerative hierarchical clustering using $z_r$ as distance. In the present application, only spatially contiguous datums (except when separated by an outlier) are merged, and the partitioned subsets correspond to segments of different log2-ratios.

## The SAD algorithm: clustering

SAD has two clustering modes: the linear mode (LM) for low-resolution arrays or when computation time is not a concern, and the parallel mode (PM) otherwise. LM has a single parameter $z_0$ while PM has an additional parameter $N_s$ whose default value of 100 is highly recommended. The steps in LM are: (i) Computation of $\tilde{\sigma}$. Let $\{v_i | i = 1,N\}$ be the initial data of log2-ratio, $q_i = v_{i+1} - v_i$ and $SD_q$ be the standard deviation of the $q_i$'s, then

$$\tilde{\sigma} = SD_q/\sqrt{2}. \tag{4}$$

$\tilde{\sigma}$ measures datum error and is sensitive only to the existence of breakpoints, which are assumed to be sparse. Treat each datum as a single-datum cluster and assign $G(y;v_i,\tilde{\sigma}^2)$ to the $i$-th datum-cluster. (ii) Selection of $z_0$. This stipulates when PGM iteration stops and addresses the statistical issues discussed in the following subsection. (iii) PGM Phase I. Perform chromosome-wide PGM iteratively to all contiguous cluster pairs. At the end of this phase each remaining single-datum cluster is a 'loner' whose existence prevents the merging of its two neighbouring clusters even if they are resolvable. (iv) PGM Phase II. Along with contiguous pairs, continue step (iii) to merge loner-divided pairs. After a loner-divided pair is merged the dividing loner becomes an 'outlier' and is excluded from subsequent calculation. At the end of this stage each of the resultant clusters is a 'segment' with an associated Gaussian $G(y;\mu,\sigma^2)$ serving as a PDF for its true value. (v) Normalization. Perform genome-wide PGM on the entire set of segments to merge contiguous as well as unconnected segment pairs. Identify the largest resultant cluster and denote its mean by $\tilde{\mu}$, here called the 'baseline'. The baseline will be taken as the reference for CNV significance test.

As PGM involves very little computation, LM is inherently a fast algorithm. On the other hand, owing to the iterative procedure, the problem size is $\mathcal{O}(N^2)$, implying long computation time when $N$ is large. PM reduces the problem size to $\mathcal{O}(N)$ with little sacrifice in accuracy. In that case, a sampling size $N_s$ is selected (by the user) and the various steps in LM are adjusted as follows. In (v), $\tilde{\mu}$ is computed using only the widest $N_s$ segments. This reduces problem size from $\mathcal{O}(N_{seg}^2)$, where $N_{seg}$ is the number of resultant segments, to $\mathcal{O}(N_s^2)$. In (iii) and (iv), prior to merging the entire current cluster set is partitioned to subsets of $N_s$ contiguous clusters, plus a remainder. The subsets are processed in parallel and the most unresolvable pair in each subset, if there is any, is merged. Thereafter the subsets of clusters (some of which have been reduced in size through merging) are joined with the remainder circularly, with the beginning of the remainder taken as the starting point, and readied for a new round of partition and merging. This is a dynamical procedure resulting in a different partition in each iteration. The problem size for each of the $N/N_s$ subsets is $\mathcal{O}(N_s^2)$, making the total problem size $\mathcal{O}(NN_s)$.

### The SAD algorithm: CNV calling and selection of $z_0$

After clustering, consider two contiguous segments: a narrow segment $s_1$ of $G_1 = G(y;\mu_1,\tilde{\sigma}^2/w_1)$ and a much wider non-CNV segment $s_2$ of $G_2 = G(y;\tilde{\mu},\tilde{\sigma}^2/w_2)$. Let $H_a$ be the null hypothesis that $s_1$ is non-CNV (i.e. the true value of $s_1$ is $\tilde{\mu}$). An independent one-sample $z$-test using a $z$-value, here called the 'aberrance',

$$z_a(G_1) \equiv (\mu_1 - \tilde{\mu})\sqrt{w_1}/\tilde{\sigma}, \qquad \mathbf{(5)}$$

yields a $P$-value for testing $H_a$, as is expected by the central limit theorem. From Equations (3 and 5), because $w_2 \gg w_1$, we have

$$z_r(G_1,G_2) \approx (\mu_1 - \tilde{\mu})\sqrt{w_1}/\tilde{\sigma} = z_a(G_1). \qquad \mathbf{(6)}$$

The lower bound for $|z_r(G_1,G_2)|$, $z_0$, is therefore also the approximate lower bound for $|z_a(G_1)|$. We therefore employ $p_0$, the corresponding $P$-value of $z_0$, as the significance level for testing $H_a$. We call $s_1$ a CNV if $|z_a(G_1)| \geq z_0$. More specifically, we call the segment a 'gain' if $z_a(G_1) \geq z_0$, or a 'loss' if $z_a(G_1) \leq -z_0$.

Because $|\mu_1 - \tilde{\mu}|/\tilde{\sigma}$ is just the signal to noise ratio (SNR) of $s_1$, Equation (6) leads to

$$\sqrt{w_1} \gtrsim z_0/\mathrm{SNR}. \qquad \mathbf{(7)}$$

That is, if SNR is known, $z_0$ also sets an approximate lower bound for CNV width.

### Software availability

The SAD program is available for download at: http://www.sybbi.ncu.edu.tw/software.htm or upon request by email at: pairwise.gaussian.merging@gmail.com.

## RESULTS

In Lai *et al.* (11) (hereafter referred to as LJKP) the performances of 11 CNV algorithms—3 smoothing-only (SO) algorithms, lowess, wavelet (12) and quantreg (13) and 8 estimation-performing (EP) algorithms, CGHseg (14), CBS (15), ChARM (16), ACE (17), HMM (18), GLAD (19), GA (20) and CLAC (21)—were compared using simulated data for testing receiver operating characteristic (ROC) as well as real Glioblastoma Multiforme (GBM) data. LJKP found that the overall top three EP performers were CGHseg, CBS and GLAD. In Fiegler *et al.* (22) two more recently developed EP algorithms, CNVfinder (22) and SW-ARRAY (23), were compared in accuracy using real data. Among these algorithms only CALC and ACE provide quantitative statistics.

We test SAD against the 10 EP algorithms in ROC. The SO algorithms were excluded because they do not explicitly address breakpoints. The ones rated accurate, CGHseg, CBS and GLAD, were further compared to SAD in speed and memory. In addition we validated SAD on low- and high-resolution data sets. We designate a SAD run in LM by SAD($z_0$,–) and in PM by SAD($z_0$,$N_s$).

### Accuracy

We calculated (details in Supplementary Data) the ROC curves of SAD the same way as in LJKP except that for better statistics we generated 10 000 instead of 100 simulated chromosomes (of 100 datums each) for each parameter set in each setting. The results (Supplementary Figure S1) indicate that a higher $z_0$ is more suitable for easy settings (wide CNV and large SNR) while a lower $z_0$ better facilitates CNV detection in difficult settings (narrow CNV or small SNR). Table 1 compares SAD($z_0$,100), $z_0 = 1.5$, 2.0 and 4.0, in area-under-curve

**Table 1.** Comparison in AUC value of ROC, of SAD against existing algorithms for two easy settings, (SNR,width) = (4,20) and (3,40), and two difficult settings, (2,5) and (1,10)

| Algorithm | (4,20) | (3,40) | (2,5) | (1,10) |
|---|---|---|---|---|
| SAD(1.5,100) | 0.99 | 0.99 | 0.93 | 0.83 |
| SAD(2.0,100) | 0.99 | 0.99 | 0.92 | 0.84 |
| SAD(4.0,100) | 0.99 | 0.99 | 0.71 | 0.59 |
| ACE | 0.99 | 0.92 | 0.73 | 0.57 |
| CBS | 0.99 | 0.99 | 0.75 | 0.59 |
| CGHseg | 0.99 | 0.99 | 0.94 | 0.78 |
| ChARM | 0.93 | 0.91 | 0.50 | 0.50 |
| CLAC | 0.97 | 0.95 | 0.84 | 0.68 |
| GA | 0.99 | 0.99 | 0.55 | 0.51 |
| GLAD | 0.99 | 0.99 | 0.56 | 0.51 |
| HMM | 0.99 | 0.99 | 0.65 | 0.54 |
| SW-ARRAY | 0.86 | 0.82 | 0.53 | 0.52 |
| CNVfinder | 0.97 | 0.95 | 0.90 | 0.75 |

(AUC) value with the 10 EP algorithms for two easy settings, (SNR,width) = (4,20) and (3,40), and two difficult settings, (2,5) and (1,10). Numbers for the eight LJKP-tested algorithms were read from Figure 2 in LJKP. Numbers for SW-ARRAY and CNVfinder were calculated using their reportedly optimal parameter values. In the easy settings, SAD(1.5–4.0,100), CBS, CGHseg, GA, GLAD and HMM perform well. In the difficult settings, SAD(1.5–2.0,100) is the best performer and CGHseg is next. Although CNVfinder performs above average in the difficult settings, it is below average in the easy settings.

In PM, higher computation speed is facilitated by using a smaller $N_s$. Because PM alters the clustering order relative to that in LM, this can induce error when $N_s$ is too small. We tested SAD in this regard and find that overall error is negligible when $N_s \gtrsim 100$ (Supplementary Figure S2).

### Speed and memory

All calculations reported here were carried out on a computer with Intel Core 2 Duo T7500 2.2G (L2:4M) CPU, 2GBs of DDRII memory, and uses Windows XP as operating system. All programs ran as a single thread and uses 50% of the CPU. Our SAD program is written in Visual C++. The other algorithms were tested with provided programs at default parameter values. The simulated chromosomes were generated with SNR = 2. Each simulated chromosome had either one or two gains. For planting the gains each chromosome was divided into five same-width sections. The second section was amplified in one-gain cases, and the second and the forth sections were amplified in two-gain cases. Computation time $\tau$ was measured for each case; the difference in $\tau$ between one and two gains reflects the dependence of speed on genomic profiles. Memory test was read from the processes tab of Windows Task Manager and involves two steps: data loading and data processing. The reading between the two steps, denoted by $\kappa_d$, is memory used for program and data. The maximum reading during data processing was recorded as $\kappa_o$ and the difference $\kappa_p = \kappa_o - \kappa_d$ was taken to be the maximum memory needed for data processing. The power-law exponents $\gamma_\tau$ and $\gamma_\kappa$ were derived from the $N$ dependences of $\tau$ and $\kappa_p$, respectively.

We compared SAD(10,100) to CGHseg, CBS and GLAD and show the results in Figure 4. We see that: (i) SAD is vastly faster than the others; at $N \approx 10^6$ it is already two orders of magnitude faster than CBS, its closest competitor. (ii) In computation time SAD is $\mathcal{O}(N)$ while GLAD and CGHseg are $\mathcal{O}(N^2)$. CBS, claimed to be $\mathcal{O}(N)$ at low resolution (24), becomes $\mathcal{O}(N^2)$ at $N \approx 5 \times 10^5$. (iii) Speed dependence on genomic profile, reflected by the difference between the 1-gain results and the 2-gain results, is significant for CBS, minor for GLAD and CGHseg, and negligible for SAD. (iv) SAD requires the least amount of memory, overall ($\kappa_o$) as well as for data-processing ($\kappa_p$). (v) In memory requirement SAD and GLAD scale as $\mathcal{O}(N)$, CBS displays irregularity, and CGHseg scales as $\mathcal{O}(N^2)$. On a computer with 2 GBs of memory, CGHseg ceases to function when $N$ exceeds about 16 000. For this reason CGHseg is not considered for further comparison.

Using real data, we ran SAD(10,100) on a 1.8-million-probeset Affymetrix Genome-Wide Human SNP Array 6.0 hybridized with a colorectal cancer sample, and measured $\tau = 8$ seconds and $\kappa_o = 323$ MBs.

### Validation on a low-resolution data set

We used a 2276-BAC public data set from the NIGMS Human Genetics Cell Repository (25) (henceforth the Snijders dataset) to perform low-resolution validation of SAD and to demonstrate the utility of $z_0$ for limiting CNV width. The dataset corresponds to 15 human cell strains. As identified by spectral karyotyping, each cell strain has either one or two CNVs and eight of the CNVs on six strains were detected to be whole-chromosome. We set a value of $z_0$ using Equation (7). For trisomic segments, the data set has SNR $\approx 0.58/0.09$, where $0.58 \approx \log_2(3/2)$ is approximately the log2-ratio of a trisomic segment and 0.09 is the value for $\tilde{\sigma}$ obtained from Equation (4). To detect a minimum CNV width between one datum (because one-datum CNVs are likely to be outliers) and two, $6.4 < z_0 < 9.1$ is required. We therefore used SAD(8,100) for this calculation.

Because the data set had previously been examined by GLAD (19) and CBS (15), we compared the three sets of results in full details in Supplementary Table S1, and summarize the comparison as follows. (1) SAD(8,100) detects more CNVs than GLAD and CBS do. (2) SAD(8,100)

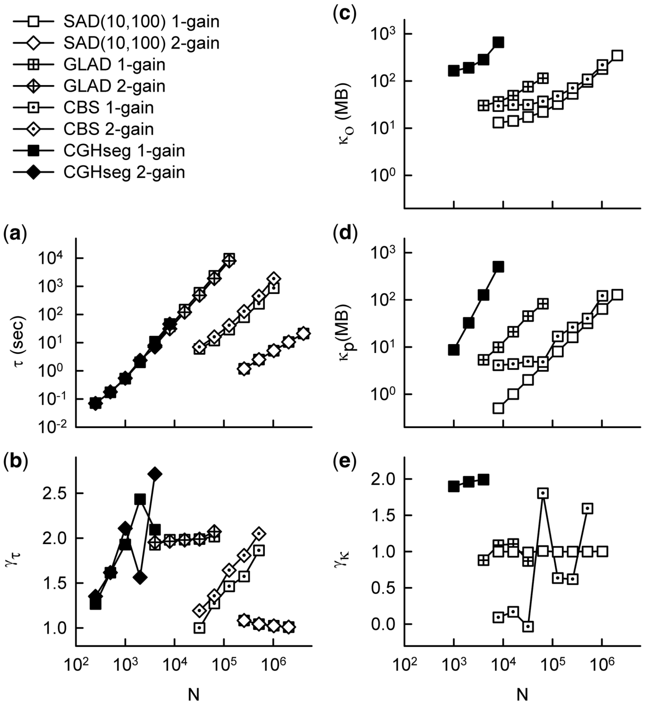

**Figure 4.** Comparisons of SAD to CGHseg, CBS and GLAD in speed and memory requirement. (**a**) Computation time $\tau$ versus $N$. (**b**) Power-law exponent $\gamma_\tau$ for $\tau$ derived from (a). (**c**) Overall memory $\kappa_o$ versus $N$. (**d**) Data-processing memory $\kappa_p$ versus $N$. (**e**) Power-law exponents $\gamma_\kappa$ for $\kappa_p$ derived from (d).

gives far fewer false-positives; the average numbers of false positive breakpoints per cell strain are 2/15, 46/15, 26/15, 37/9 and 16/9 for SAD(8,100), GLAD($\lambda' = 8$), GLAD($\lambda' = 10$), CBS($\alpha = 0.01$) and CBS($\alpha = 0.001$), respectively. (3) SAD alone assigns a $z$-value to each CNV for assessing significance. (4) SAD(8,100) alone detects whole-chromosome CNVs on whose detection GLAD and CBS are silent because they are based on breakpoint detection within chromosomes.

### Validation on a high-resolution dataset

In Redon *et al.* (3), 43 genomic regions were examined by SYBR real-time PCR or MassSpec to validate the respective CNV calls for NA15510 vs NA10851 on the Affymetrix 500K EA platform. We used three of these regions, cnp8, cnp23 and cnp36, respectively determined in (3) to be gain, loss and gain, to validate SAD and to demonstrate the utility of $z_0$ for characterizing CNV significance. In Figure 5, the results of three runs, SAD(10,100), SAD(8,100) and SAD(6,100), on the first Sty replicates of the Redon dataset are respectively shown in frame sets (a), (b) and (c). At $z_0 = 10$ (Figure 5a) only cnp36 is detected with $z_a = 10.6$. When $z_0$ is lowered to 8 (Figure 5b), cnp23 is detected with $z_a = -8.2$. When $z_0$ is further lowered to 6 (Figure 5c), cnp8 is detected with $z_a = 7.4$.

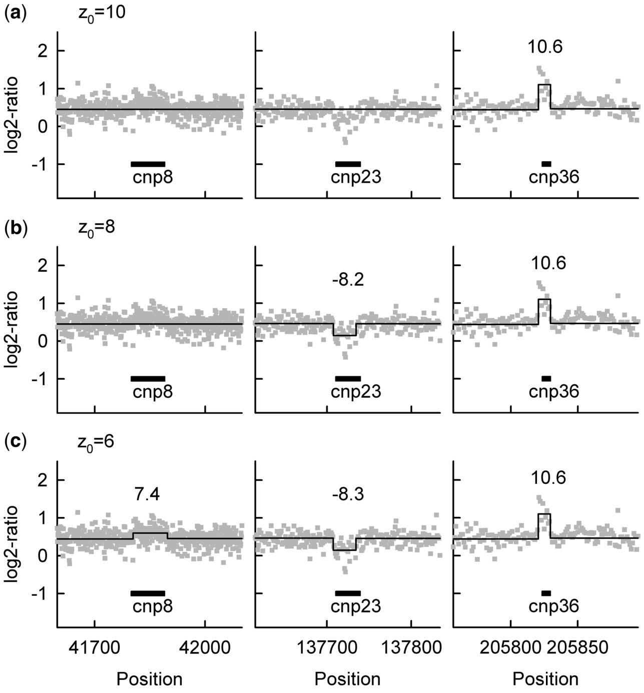

**Figure 5.** A high-resolution validation test for SAD on 3 genomic regions with known CNVs, whose positions are shown as thick black segments in the frames. The three sets of frames are for the three runs: (**a**) SAD(10,100); (**b**) SAD(8,100); and (**c**) SAD(6,100). Data and SAD predictions are respectively shown as solid grey squares and black lines. Shown above each CNV detected by SAD is its aberrance $z_a$.

## DISCUSSION

We have demonstrated that by virtue of its accuracy, parsimony in memory use and speed, SAD can manage the challenges analyzing modern high-resolution microarrays significantly better than existing algorithms. Algorithmically SAD is easy to understand because it employs fundamental principles of statistics and precise but very simple mathematics [as compared to the mathematics in the formulation of, say, GLAD (19)]. SAD makes all internal decisions based on statistics and provides an external quantitative statistic. With only two user-tunable parameters, $z_0$ and $N_s$, the meanings of which are both intuitively accessible, SAD is also the easiest to use. Users can select $z_0$, the primary parameter, based on their requirement for CNV significance or CNV width. We recommend setting the second parameter, $N_s$, to 100. This guarantees good accuracy and a computation time that is $\mathcal{O}(N)$.

Quantitative statistics provide the basis on which a level of confidence may be assigned to each inference and for setting a priority for experimental confirmation for such inferences. All measurements, especially those involving microarrays, carry inherent statistical error. SAD quantifies such errors as data uncertainty, tracks the latter throughout a clustering process using exact mathematical relations, and provides $z$-values for assessing CNV significance. The $z$-values, when used for downstream calculations such as the identification of recurrent aberrations using multiple arrays, allows the initial uncertainty to be passed on further.

SAD is an application build on PGM. The upgrading of SAD computation time from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ is a consequence of the parallel processing made possible by the employment of agglomerative hierarchical clustering in PGM. The superior accuracy of SAD results from the exploitation by PGM of a common trait seen in most systems: that measurement errors are normally distributed. The operating principle of SAD is accessible to the user because in PGM the resolving power used for determining breakpoints is controlled via an intuitive statistic threshold. These properties of PGM promise its usefulness and wide application, beyond CNV, in the general analysis of microarray data.

## SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

## FUNDING

National Science Council (ROC) (Grant 97-2112-M-008-013; in part); Cathay General Hospital-NCU Collaboration (Grant 97-CGH-NCU-A1, in part) Funding for open access charge: National Science Council and the Ministry of Education (Research grants to H.C.L.).

*Conflict of interest statement*. None declared.

## REFERENCES

1. Pollack,J.R., Perou,C.M., Alizadeh,A.A., Eisen,M.B., Pergamenschikov,A., Williams,C.F., Jeffrey,S.S., Botstein,D. and Brown,P.O. (1999) Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nat. Genet.*, **23**, 41–46.
2. Sebat,J., Lakshmi,B., Troge,J., Alexander,J., Young,J., Lundin,P., Månér,S., Massa,H., Walker,M., Chi,M. *et al.* (2004) Large-Scale copy number polymorphism in the human genome. *Science*, **305**, 525–528.
3. Redon,R., Ishikawa,S., Fitch,K.R., Feuk,L., Perry,G.H., Andrews,T.D., Fiegler,H., Shapero,M.H., Carson,A.R., Chen,W. *et al.* (2006) Global variation in copy number in the human genome. *Nature*, **444**, 444–454.
4. Barnes,C., Plagnol,V., Fitzgerald,T., Redon,R., Marchini,J., Clayton,D. and Hurles,M.E. (2008) A robust statistical method for case-control association testing with copy number variation. *Nat. Genet.*, **40**, 1245–1252.
5. Solinas-Toldo,S., Lampel,S., Stilgenbauer,S., Nickolenko,J., Benner,A., Dohner,H., Cremer,T. and Lichter,P. (1997) Matrix-based comparative genomic hybridization: biochips to screen for genomic imbalances. *Gene Chromosome. Canc.*, **20**, 399–407.
6. Pinkel,D., Segraves,R., Sudar,D., Clark,S., Poole,I., Kowbel,D., Collins,C., Kuo,W.L., Chen,C., Zhai,Y. *et al.* (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.*, **20**, 207–211.
7. Pinkel,D. and Albertson,D.G. (2005) Array comparative genomic hybridization and its applications in cancer. *Nat. Genet.*, **37(Suppl.)**, 11–17.
8. Brennan,C., Zhang,Y., Leo,C., Feng,B., Cauwels,C., Aguirre,A.J., Kim,M., Protopopov,A. and Chin,L. (2004) High-resolution global profiling of genomic alterations with long oligonucleotide microarray. *Cancer Res.*, **64**, 4744–4748.
9. Lucito,R., Healy,J., Alexander,J., Reiner,A., Esposito,D., Chi,M., Rodgers,L., Brady,A., Sebat,J., Troge,J. *et al.* (2003) Representational oligonucleotide microarray analysis: a highresolution method to detect genome copy number variation. *Genome Res.*, **13**, 2291–2305.
10. Ishkanian,A.S., Malloff,C.A., Watson,S.K., deLeeuw,R.J., Chi,B., Coe,B.P., Snijders,A., Albertson,D.G., Pinkel,D., Marra,M.A. *et al.* (2004) A tiling resolution DNAmicroarray with complete coverage of the human genome. *Nat. Genet.*, **36**, 299–303.
11. Lai,W.R., Johnson,M.D., Kucherlapati,R. and Park,P.J. (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, **21**, 3763–3770.
12. Hsu,L., Self,S.G., Grove,D., Randolph,T., Want,K., Delrow,J.J., Loo,L. and Porter,P. (2005) Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, **6**, 211–226.
13. Eilers,P.H.C. and de Menezes,R.X. (2005) Quantile smoothing of array CGH data. *Bioinformatics*, **21**, 1146–1153.
14. Picard,F., Robin,S., Lavielle,M., Vaisse,C. and Daudin,J. (2005) A statistical approach for array CGH data analysis. *BMC Bioinforma.*, **6**, 27.
15. Olshen,A.B., Venkatraman,E.S., Lucito,R. and Wigler,M. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.
16. Myers,C.L., Dunham,M.J., Kung,S.Y. and Troyanskaya,O.G. (2004) Accurate detection of aneuploidies in array CGH and gene expression microarray data. *Bioinformatics*, **20**, 3533–3543.
17. Lingjærde,O.C., Baumbusch,L.O., Liestøl,K., Glad,I.K. and Børresen-Dale,A. (2005) CGH-Explorer: a program for analysis of array-CGH data. *Bioinformatics*, **21**, 821–822.
18. Fridlyand,J., Snijders,A.M., Pinkel,D., Albertson,D.G. and Jain,A.N. (2004) Hidden Markov models approach to the analysis of array CGH data. *J. Multivariate Anal.*, **90**, 132–153.
19. Hupé,P., Stransky,N., Thiery,J., Radvanyi,F. and Barillot,E. (2004) Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, **20**, 3413–3422.
20. Jong,K., Marchiori,E., van der Vaart,A., Ylstra,B., Weiss,M. and Meijer,G. (2003) Chromosomal breakpoint detection in human cancer. In *Lecture Notes in Computer Science*. Springer, Berlin, pp. 54–65.
21. Wang,P., Kim,Y., Pollack,J., Narasimhan,B. and Tibshirani,R. (2005) A method for calling gains and losses in array CGH data. *Biostatistics*, **6**, 45–58.
22. Fiegler,H., Redon,R., Andrews,D., Scott,C., Andrews,R., Carder,C., Clark,R., Dovey,O., Ellis,P., Feuk,L. *et al.* (2006) Accurate and reliable high-throughput detection of copy number variation in the human genome. *Genome Res.*, **16**, 1566–1574.
23. Price,T.S., Regan,R., Mott,R., Hedman.Å, Honey,B., Daniels,R.J., Smith,L., Gerrnfield,A., Tiganescu,A., Buckle,Vl. *et al.* (2005) SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. *Nucleic Acids Res.*, **33**, 3455–3464.
24. Venkatraman,E.S. and Olshen,A.B. (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, **23**, 657–663.
25. Snijders,A.M., Nowak,N., Segraves,R., Blackwood,S., Brown,N., Conroy,J., Hamilton,G., Hindle,A.K., Huey,B., Kimura,K. *et al.* (2001) Assembly of microarrays for genome-wide measurement of DNA copy number. *Nat. Genet.*, **29**, 263–264.