JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 13, Number 2, 2006

Pp. 215–228

Efficient Calculation of Interval Scores for DNA Copy Number Data Analysis

DORON LIPSON,[1] YONATAN AUMANN,[2] AMIR BEN-DOR,[3] NATHAN LINIAL,[4] and ZOHAR YAKHINI[1,3]

ABSTRACT

DNA amplifications and deletions characterize cancer genome and are often related to disease evolution. Microarray-based techniques for measuring these DNA copy-number changes use fluorescence ratios at arrayed DNA elements (BACs, cDNA, or oligonucleotides) to provide signals at high resolution, in terms of genomic locations. These data are then further analyzed to map aberrations and boundaries and identify biologically significant structures. We develop a statistical framework that enables the casting of several DNA copy number data analysis questions as optimization problems over real-valued vectors of signals. The simplest form of the optimization problem seeks to maximize $\varphi(I) = \sum v_i/\sqrt{|I|}$ over all subintervals I in the input vector. We present and prove a linear time approximation scheme for this problem, namely, a process with time complexity $O\left(n\epsilon^{-2}\right)$ that outputs an interval for which $\varphi(I)$ is at least $\text{Opt}/\alpha(\epsilon)$, where Opt is the actual optimum and $\alpha(\epsilon) \to 1$ as $\epsilon \to 0$. We further develop practical implementations that improve the performance of the naive quadratic approach by orders of magnitude. We discuss properties of optimal intervals and how they apply to the algorithm performance. We benchmark our algorithms on synthetic as well as publicly available DNA copy number data. We demonstrate the use of these methods for identifying aberrations in single samples as well as common alterations in fixed sets and subsets of breast cancer samples.

Key words: CGH, cancer, microarray analysis, optimization, approximation.

1. INTRODUCTION

Alterations in DNA copy number are characteristic of many cancer types and are thought to drive some cancer pathogenesis processes. These alterations include large chromosomal gains and losses as well as smaller scale amplifications and deletions. Because of their role in cancer development, regions of chromosomal instability are useful for elucidating other components of the process. For example, since genomic instability can trigger the overexpression or activation of oncogenes and the silencing of tumor

[1]Computer Science Department, Technion, Haifa 32000, Israel.
[2]Computer Science Department, Bar-Ilan University, Ramat Gan 52900, Israel.
[3]Agilent Technologies Israel, 94 Em Hamoshavot, Petach-Tikva 49527, Israel.
[4]Computer Science Department, Hebrew University, Givat Ram, Jerusalem 91904, Israel.

suppressors, mapping regions of common genomic aberrations has been used to discover cancer-related genes. Understanding genome aberrations is important for both the basic understanding of cancer and for diagnosis and clinical practice.

Alterations in DNA copy number have been initially measured using local fluorescence in situ hybridization-based techniques. These evolved to a genomewide technique called comparative genomic hybridization (CGH, see Kallioniemi *et al.* [1993]), now commonly used for the identification of chromosomal alterations in cancer (Mertens *et al.*, 1997; Balsara and Testa, 2002). In this genomewide cytogenetic method, differentially labeled tumor and normal DNA are co-hybridized to normal metaphases. Ratios between the two labels allow the detection of chromosomal amplifications and deletions of regions that may harbor oncogenes and tumor suppressor genes. Classical CGH has, however, a limited resolution (10–20 Mbp). With such low resolution, it is impossible to predict the borders of the chromosomal changes or to identify changes in copy numbers of single genes and small genomic regions. In a more advanced method, termed array CGH (aCGH), tumor, and normal DNA are co-hybridized to a microarray of thousands of genomic clones of BAC, cDNA, or oligonucleotide probes (Pollack *et al.*, 1999; Hyman *et al.*, 2002; Pinkel *et al.*, 1998; Hedenfalk *et al.*, 2003; Sebat *et al.*, 2004; Brennan *et al.*, 2004; Bignell *et al.*, 2004). The use of aCGH allows the determination of changes in DNA copy number of relatively small chromosomal regions. Using oligonucleotides arrays, the resolution can, in theory, be finer than single genes.

To fully realize the advantages associated with the emerging high resolution technologies, practitioners need appropriate efficient data analysis methods. A common first step in analyzing DNA copy number (DCN) data consists of identifying aberrant (amplified or deleted) regions in each individual sample. Indeed, current literature on analyzing DCN data describes several approaches to this task, based on a variety of optimization techniques. Hupe *et al.* (2004) develop a methodology for automatic detection of breakpoints and aberrant regions, based on adaptive weight smoothing (Polzehl and Spokoiny, 2000), a segmentation technique that fits a piecewise constant function to an input function. The fit is based on maximizing the likelihood of the (observed) function given the piecewise constant model, penalized for the number of transitions. The penalty weight is a parameter of the process. Sebat *et al.* (2004), in a pioneering paper, study DCN variations that naturally occur in normal populations. Using an HMM-based approach, they compare signals for two individuals and seek intervals of four or more probes in which DCNs are likely to be different. A common shortcoming of many other current approaches is lack of principled optimization criteria that drive the method. Even when a figure of merit forms the mathematical basis of the process such as in Sebat *et al.* (2004), convergence of the optimization process is not guaranteed.

Further steps in analyzing DCN data include the automatic elucidation of more complex structures. A central task involves the discovery of common aberrations, either in a fixed set of studied samples or in an unsupervised mode, where we search for intervals that are aberrant in some significant subset of the samples. There are no formal treatments of this problem in the literature, but most studies do report common aberrations and their locations. The relationship of DCN variation and expression levels of genes that reside in the aberrant region is of considerable interest. By measuring DNA copy numbers and mRNA expression levels on the same set of samples, we gain access to the relationship of copy number alterations to how they are manifested in altering expression profiles. In Platzer *et al.* (2002) the authors used (metaphase slides) CGH to identify large-scale amplifications in 23 metastatic colon cancer samples. For each identified large amplified region, they compared the median expression levels of genes that reside there (2,146 genes total), in the samples where amplification was detected, to the median expression levels of these genes in 9 normal control colon samples. A two-fold overexpression was found in 81 of these genes. No quantitative statistical assessment of the results is given. In Pollack *et al.* (2002) a more decisive observation is reported. For breast cancer samples, the authors establish a strong global correlation between copy number changes and expression level variation. Hyman *et al.* (2002) report similar findings. The statistics used by both latter studies is based on simulations and takes into account single gene correlations but not local regional effects. In Section 2.3, we show how our general framework enables us to efficiently compute correlations between gene expression vectors and DCN vectors of genomic intervals, greatly extending the scope of the analysis.

In summary, current literature on CGH data analysis does not address several important aspects of DCN data analysis and addresses others in an informal mathematical setup. In particular, the methods described in current literature do not directly address the tasks of identifying aberrations that are common to a significant subset of a study sample set. In Section 2, we present methods that optimize a clear, statistically motivated, score function for genomic intervals. The analysis then reports all high-scoring intervals as candidate

aberrant regions. Our algorithmic approach yields performance guarantees as described in Sections 3 and 4. The methods can be used to automatically map aberration boundaries as well as to identify aberrations in subsets and correlations with gene expression, all within the same mathematical framework. Actual results from analyzing DCN data are described in Section 5.

Our approach is based on finding intervals of consistent high or low signals within an ordered set of signals, coming from measuring a set of genomic locations and considered in their genomic order. This principle motivates us to assign scores to intervals I of signals. The scores are designed to reflect the statistical significance of the observed consistency of high or low signals. These interval scores are useful in many levels of the analysis of DCN data. By using adequately defined statistical scores, we transform the task of suggesting significant common aberrations as well as other tasks to optimizing segment scores in real-valued vectors. Segment scores are also useful in interpreting other types of genomic data, such as LOD scores in genetic analysis (Morley *et al.*, 2004).

The computational problem of optimizing interval scores for vectors of real numbers is related to segmentation problems, widely used in time series analysis as well as in image processing (Himberg *et al.*, 2001; Fu and Mui, 1981).

2. INTERVAL SCORES FOR CGH

In this section, we formally define interval scores for identifying aberrant chromosomal intervals using DCN data. In Section 2.1, we define a basic score that is used to identify aberrations in a single sample. In Sections 2.2 and 2.3, we extend the score to accommodate data from multiple samples as well as joint DCN and gene-expression data.

2.1. Aberrant intervals in single samples

Detection of chromosomal aberrations in a single sample is performed for each chromosome separately. Let $V = (v_1, \ldots, v_n)$ denote a vector of DCN data for one chromosome (or chromosome arm) of a single sample, where v_i denotes the (normalized) data for the i-th probe along the chromosome. The underlying model of chromosomal instabilities suggests that amplification and deletion events typically span several probes along the chromosome. Therefore, if the target chromosome contains amplification or deletion events, then we expect to see many consecutive positive entries in V (amplification), or many consecutive negative entries (deletion). On the other hand, if the target chromosome is normal (no aberration), we expect no localized effects. Intuitively, we look for intervals (sets of consecutive probes) where signal sums are significantly larger or significantly smaller than expected at random. To formalize this intuition, we assume (null model) that there is no aberration present in the target DNA and therefore the variation in V represents only the noise of the measurement.

Assuming that the measurement noise along the chromosome is independent for distinct probes and normally distributed,[1] let μ and σ denote the mean and standard deviation of the normal genomic data (typically, after normalization $\mu = 0$). Given an interval I spanning $|I|$ probes, let

$$\varphi^{sig}(I) = \sum_{i \in I} \frac{(v_i - \mu)}{\sigma \sqrt{|I|}}. \tag{1}$$

Under the null model, $\varphi^{sig}(I)$ has a Normal$(0, 1)$ distribution, for any I. Thus, We can use $\varphi^{sig}(I)$ (which does not depend on probe density) to assess the statistical significance of values in I using, for example, the following large deviation bound (Feller, 1970):

$$\text{Prob}(|\varphi^{sig}(I)| > z) \approx \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{z} e^{-\frac{1}{2}z^2}. \tag{2}$$

Given a vector of measured DCN data V, we therefore seek all intervals I with $\varphi^{sig}(I)$ exceeding a certain threshold. Setting the threshold to avoid false positives, we report all these intervals as putative aberrations.

[1]The normality assumption of the noise can be somewhat relaxed as the distribution of average noise for large intervals will, in any event, be close to normal (central limit theorem).

2.2. Aberrant intervals in multiple samples

When DCN data from multiple samples is available, it may, of course, be processed one sample at a time. However, it is possible to take advantage of the additional data to increase the significance of the located aberrations by searching for *common* aberrations. More important than the statistical advantage, common aberrations are indicative of genomic alterations selected for in the tumor development process and may therefore be of higher biological relevance.

Given a set of samples S and a matrix $V = \{v_{s,i}\}$ of CGH values where $v_{s,i}$ denotes the (normalized) data for the i-th probe in sample $s \in S$, identification of common aberrations may be done in one of two modes. In the *fixed set* mode, we search for genomic intervals that are significantly aberrant (either amplified or deleted) in all samples in S. In the *class discovery* mode, we search for genomic intervals I for which there exists a subset of the samples $C \subseteq S$ such that I is significantly aberrant only in the samples within C.

Fixed set of samples. A simple variant of the single-sample score (1) allows accommodation of multiple samples. Given an interval I, let

$$\varphi_S^{sig}(I) = \sum_{s\in S}\sum_{i\in I}\frac{(v_{s,i}-\mu)}{\sigma\sqrt{|I|\cdot|S|}}. \tag{3}$$

Although this score will indeed indicate whether I contains a significant aberration within the samples in S, it does not have the ability to discern between an *uncommon* aberration that is highly manifested in only one sample and a *common* aberration that has a more moderate effect on all samples. In order to focus on common aberrations, we employ a *robust* variant of the score: Given some threshold τ^+, we create a binary dataset: $B = \{b_{s,i}\}$ where $b_{s,i} = 1$ if $v_{s,i} > \tau^+$ and $b_{s,i} = 0$ otherwise. Assuming (null model) that the appearance of 1s in B is independent for distinct probes and samples, we expect the number of positive values in the submatrix defined by $I \times S$ to be Binom(n, p) distributed with $n = |I| \cdot |S|$ and $p = \text{Prob}(v_{s,i} > \tau^+) = \sum_{s\in S}\sum_{i=1}^{n}\frac{b_{s,i}}{|B|}$. The significance of k 1s in $I \times S$ can be assessed by the binomial tail probability:

$$\sum_{i=k}^{n}\binom{n}{i}p^i(1-p)^{(n-i)}. \tag{4}$$

For algorithmic convenience, we utilize the Normal approximation of the above (DeGroot, 1989). Namely, for each probe i we define the score of an interval I as in (3):

$$\varphi_S^{rob}(I) = \sum_{s\in S}\sum_{i\in I}\frac{(b_{s,i}-\nu)}{\rho\sqrt{|I|\cdot|S|}}, \tag{5}$$

where $\nu = p$ and $\rho = \sqrt{p(1-p)}$ are the mean and standard deviation of the variables $b_{s,i}$. A high score $\varphi_S^{rob}(I)$ is indicative of a common amplification in I. A similar score indicates deletions using a negative threshold τ^-.

Class discovery. In the mode of class discovery, we search a genomic interval I and a subset $C \subseteq S$ such that I is significantly aberrant on the samples within C. Formally, we search for a pair (I, C) that maximizes the score

$$\varphi^{sig}(I, C) = \sum_{s\in C}\sum_{i\in I}\frac{(v_{s,i}-\mu)}{\sigma\sqrt{|I|\cdot|C|}}. \tag{6}$$

A *robust* form of this score is

$$\varphi^{rob}(I, C) = \sum_{s\in C}\sum_{i\in I}\frac{(b_{s,i}-\nu)}{\rho\sqrt{|I|\cdot|C|}}. \tag{7}$$

2.3. Regional correlation to gene expression

In previous work (Lipson *et al.*, 2004), we introduced the regional correlation score as a measure of correlation between the expression levels pattern of a gene and an aberration in or close to its genomic

locus. For any given gene g with a known genomic location, the goal is to find whether there is an aberration in its chromosome that potentially effects the transcription levels of g. Formally, we are given a vector e of expression level measurements of g over a set of samples S and a set of vectors $V = (v_1, \ldots, v_n)$ of the same length corresponding to genomically ordered probes on the same chromosome, where each vector contains DCN values over the same set of samples S. For a genomic interval $I \subset [1, \ldots, n]$, we define a regional correlation score:

$$\varphi^{cor}(I, e) = \frac{\sum_{i \in I} r(e, v_i)}{\sqrt{|I|}}, \tag{8}$$

where $r(e, v_i)$ is some correlation score (e.g., Pearson correlation) between the vectors e and v_i. We are interested in determining whether there is some interval I for which $\varphi^{cor}(I, e)$ is significantly high. Note that for this decision problem it is sufficient to use an approximation process, such as described in Section 3.

2.4. A tight upper bound for the number of optimal intervals

In general, we are interested in finding the interval with the maximal score. A natural question is thus what is the maximal possible number of intervals with this score. The following theorem, proof of which appears in Appendix A, provides a tight bound on this number.

Theorem 1. *For any of the above scores, there can be n and at most n maximal intervals.*

3. APPROXIMATION SCHEME

In the previous section, we described interval scores arising from several different motivations related to the analysis DCN data. Despite the varying settings, the form of the interval scores in the different cases is similar. We are interested in finding the interval with maximal score. Clearly, this can be done by exhaustive search, checking all possible intervals. However, even for the single sample case this would take $\Theta(n^2)$ steps, which rapidly becomes time consuming, as the number of measured genomic loci grows to tens of thousands. Moreover, even for $n \approx 10^4$, $\Theta(n^2)$ does not allow for interactive data analysis, which is called for by practitioners (over one minute for 10 K probes on 40 samples; see Section 5.1). For the class discovery case, a näive solution would require an exponential $\Theta(n^2 2^{|S|})$ number of steps. Thus, we seek more efficient algorithms. In this section, we present a linear time approximation scheme. Then, based on this approximation scheme, we show how to efficiently find the actual optimal interval.

3.1. Fixed sample set

Note that the single sample case is a specific case of the fixed multiple-sample case ($|S| = 1$), on which we concentrate. For this case, there are two interval scores defined in Section 2.2. The optimization problem for both these scores, as well as the score for regional correlations of Section 2.3, can all be cast as a general optimization problem. Let $W = (w_1, \ldots, w_n)$ be a sequence of numbers. For an interval I, define

$$\varphi(I) = \frac{\sum_{i \in I} w_i}{\sqrt{|I|}}. \tag{9}$$

Setting $w_i = \frac{\sum_{s \in S}(v_{s,i} - \mu)}{\sigma \sqrt{|S|}}$ gives $\varphi(I) = \varphi_S^{sig}(I)$; setting $w_i = \frac{\sum_{s \in S}(b_{s,i} - \nu)}{\rho \sqrt{|S|}}$ gives $\varphi(I) = \varphi_S^{rob}(I)$; and setting $w_i = r(e, v_i)$ gives $\varphi(I) = \varphi^{cor}(I, e)$. Thus, we focus on the problem of optimizing $\varphi(I)$ for a general sequence W.

A geometric family of intervals. For an interval $I = [x, y]$, define $sum(I) = \sum_{i=x}^{y} w_i$. Fix $\epsilon > 0$. We define a family $\mathcal{I}$ of intervals of increasing lengths, the *geometric family,* as follows. For integral j,

let $k_j = (1+\epsilon)^j$ and $\Delta_j = \epsilon k_j$. For $j = 0, \ldots, \log_{(1+\epsilon)} n$, let

$$\mathcal{I}(j) = \left\{ [i\Delta_j, i\Delta_j + k_j - 1] : 0 \le i \le \frac{n - k_j}{\Delta_j} \right\}. \tag{10}$$

In words, $\mathcal{I}(j)$ consists of intervals of size k_j evenly spaced Δ_j apart. Set $\mathcal{I} = \cup_{j=0}^{\log_{(1+\epsilon)} n} \mathcal{I}(j)$. The following lemma shows that any interval $I \subseteq [1 \ldots n]$ contains an interval of $\mathcal{I}$ that has "almost" the same size.

Lemma 1. *Let I be an interval, and J the left-most longest interval of $\mathcal{I}$ fully contained in I. Then $\frac{|I|-|J|}{|I|} \le 2\epsilon + \epsilon^2$.*

Proof. Let j be such that $|J| = k_j$. In $\mathcal{I}$, there is an interval of size k_{j+1} every Δ_{j+1} steps. Thus, since there are no intervals of size k_{j+1} contained in I, it must be that $|I| < k_{j+1} + \Delta_{j+1}$. Therefore

$$|I| - |J| < k_{j+1} + \Delta_{j+1} - k_j = (1+\epsilon)k_j + \epsilon(1+\epsilon)k_j - k_j = (2\epsilon + \epsilon^2)k_j \le (2\epsilon + \epsilon^2)|I|. \quad \blacksquare$$

The approximation algorithm for a fixed set. The approximation algorithm simply computes scores for all $J \in \mathcal{I}$ and outputs the highest scoring one:

Algorithm 1. Approximation Algorithm—Fixed Sample Case

Input: Sequence $W = \{w_i\}$.
Output: Interval J with score approximating the optimal score.

$sum([1,0]) = 0$
for $j = 1$ to n $sum([1,j]) = sum([1,j-1]) + w_j$
Foreach $J = [x,y] \in \mathcal{I}$ $\varphi(J) = \frac{sum([1,y]) - sum([1,x-1])}{|I|^{1/2}}$
output $J_{max} = \text{argmax}_{J \in \mathcal{I}}\{\varphi(J)\}$

The approximation guarantee of the algorithm is based on the following lemma:

Lemma 2. *For $\epsilon \le 1/5$, the following holds. Let I^* be an interval with the optimal score, and let J be the left-most longest interval of $\mathcal{I}$ contained in I. Then $\varphi(J) \ge \varphi(I^*)/\alpha$, with $\alpha = (1 - \sqrt{2\epsilon(2+\epsilon)})^{-1}$.*

Proof. Set $M^* = \varphi(I^*)$. Assume by contradiction that $\varphi(J) < M^*/\alpha$. Denote $J = [u, v]$ and $I^* = [x, y]$. Define $A = [x, u-1]$ and $B = [v+1, y]$ (the segments of I^* protruding beyond J to the left and right). We have

$$\begin{aligned}
M^* = \varphi(I^*) &= \frac{sum(A) + sum(J) + sum(B)}{\sqrt{|I^*|}} \\
&= \frac{sum(A)}{\sqrt{|A|}} \cdot \frac{\sqrt{|A|}}{\sqrt{|I^*|}} + \frac{sum(J)}{\sqrt{|J|}} \cdot \frac{\sqrt{|J|}}{\sqrt{|I^*|}} + \frac{sum(B)}{\sqrt{|B|}} \cdot \frac{\sqrt{|B|}}{\sqrt{|I^*|}} \\
&= \varphi(A) \cdot \frac{\sqrt{|A|}}{\sqrt{|I^*|}} + \varphi(J) \cdot \frac{\sqrt{|J|}}{\sqrt{|I^*|}} + \varphi(B) \cdot \frac{\sqrt{|B|}}{\sqrt{|I^*|}} \\
&\le M^* \frac{\sqrt{|A|}}{\sqrt{|I^*|}} + \varphi(J) \frac{\sqrt{|J|}}{\sqrt{|I^*|}} + M^* \frac{\sqrt{|B|}}{\sqrt{|I^*|}} \qquad (11) \\
&\le \left(\frac{\sqrt{|A|} + \sqrt{|B|}}{\sqrt{|I^*|}} \right) M^* + \varphi(J) \le \sqrt{2 \cdot \frac{|A| + |B|}{|I^*|}} \cdot M^* + \varphi(J) \qquad (12) \\
&\le \sqrt{2(2\epsilon + \epsilon^2)} \cdot M^* + \varphi(J), \qquad (13) \\
&< \sqrt{2(2\epsilon + \epsilon^2)} \cdot M^* + M^*/\alpha = M^*, \qquad (14)
\end{aligned}$$

in contradiction. In the above, (11) follows from the optimality of M^*; (12) from arithmetic-geometric means inequality; (13) from Lemma 1; and (14) from the contradiction assumption and by the definition of α. ■

Thus, since the optimal interval must contain at least one interval of $\mathcal{I}$, we get the following.

Theorem 2. *For $\epsilon \leq 1/5$, Algorithm 1 provides an $\alpha(\epsilon) = (1 - \sqrt{2\epsilon(2+\epsilon)})^{-1}$ approximation to the maximal score.*

Note that $\alpha(\epsilon) \to 1$ as $\epsilon \to 0$. Hence, the above constitutes an approximation scheme.

Complexity. The complexity of the algorithm is determined by the number of intervals in $\mathcal{I}$. For each j, the intervals of $\mathcal{I}_j$ are Δ_j apart. Thus, $|\mathcal{I}_j| \leq \frac{n}{\Delta_j} = \frac{n}{\epsilon(1+\epsilon)^j}$. Hence, the total complexity of the algorithm is

$$|\mathcal{I}| \leq \sum_{j=0}^{\log_{(1+\epsilon)} n} \frac{n}{\epsilon}(1+\epsilon)^{-j} \leq \frac{n}{\epsilon}\sum_{j=0}^{\infty}(1+\epsilon)^{-j} = \epsilon^{-2}n = O(n\epsilon^{-2}).$$

3.2. Class discovery

Consider the problem of optimizing the scores $\varphi^{sig}(I, C)$ and $\varphi^{rob}(I, C)$. Similar to the fixed sample case, both problems can be cast as an instance of the general optimization problem. For an interval I and $C \subseteq S$, let

$$\varphi(I, C) = \frac{\sum_{i \in I, s \in C} w_{s,i}}{\sqrt{|I| \cdot |C|}}, \quad opt(I) = \max_{C \subseteq S} \varphi(I, C).$$

Note that $\max_{I,C}\{\varphi(I, C)\} = \max_I\{opt(I)\}$.

Computing *opt*(*I*). We now show how to efficiently compute $opt(I)$, without actually checking all possible subsets $C \subseteq S$. The key idea is the following. Note that for any C, $\varphi(I, C) = \sum_{s \in C} sum_s(I)/(|C||I|)^{1/2}$. Thus, for a fixed $|C| = k$, $\varphi(I, C)$ is maximized by taking k s's with the largest $sum_s(I)$. Thus, we need only sort the samples by this order, and consider the $|S|$ possible sizes, which is done in $O(|S| \log |S|)$. A description of the algorithm is provided in Appendix B.

The approximation algorithm for class discovery. Recall the geometric family of intervals $\mathcal{I}$, as defined in Section 3.1. For the approximation algorithm, for each $J \in \mathcal{I}$ compute $opt(J)$. Let $J_{\max}$ be the interval with the largest $opt(J)$. Using the Algorithm 4 find C for which $opt(J)$ is obtained and output the pair J, C. The approximation ratio obtained for the class discovery case is identical to that of the fixed case, and the analysis is also similar.

Theorem 3. *For any $\epsilon \leq 1/5$, the above algorithm provides an $\alpha(\epsilon) = (1 - \sqrt{2\epsilon(2+\epsilon)})^{-1}$ approximation to the maximal score.*

The proof, which is essentially identical to that of Theorem 2, is omitted. Again, since $\alpha(\epsilon)$ approaches 1 as ϵ approaches 0, the above constitutes an approximation scheme.

Complexity. Computing $\varphi([1, j], s)$ for $j = 1, \ldots, n$, takes $O(|S|n)$ steps. For each $J \in \mathcal{I}$ computing $opt(J)$ is $O(|S| \log |S|)$. There are $O(n\epsilon^{-2})$ intervals in $\mathcal{I}$. Thus, the total number of steps for the algorithm is $O(n|S| \log |S|\epsilon^{-2})$.

4. FINDING THE OPTIMAL INTERVAL

In the previous section, we showed how to *approximate* the optimal score. We now show how to find the absolute optimal score and interval. We present two algorithms. First, we present the *LookAhead* algorithm. Then, we present the GFA (*Geometric Family Algorithm*), which is based on a combination of the *LookAhead* algorithm, and the approximation algorithm described above. We note that for both

algorithms we cannot prove that their worst case performance is better than $O(n^2)$, but in Section 5 we show that in practice *LookAhead* run in $O(n^{1.5})$ and GFA runs in linear time.

4.1. LookAhead algorithm

Consider the fixed sample case. The algorithm operates by considering all possible interval starting points, in sequence. For each starting point i, we check the intervals with ending point in increasing distance from i. The basic idea is to try not to consider all possible ending points, but rather to skip some that will clearly not provide the optimum. Assume that we are given two parameters t and m, where t is a lower bound on the optimal score ($t \leq \max_I \varphi(I)$) and m is an upper bound on the value of any single element ($m \geq \max_i w_i$). Assume that we have just considered an interval of length k: $I = [i, \ldots, i+k-1]$ with $\sigma = sum(I)$, and $\varphi(I) = \frac{\sigma}{\sqrt{k}}$. After checking the interval I, an exhaustive algorithm would continue to the next ending point and check the interval $I_1 = [i, \ldots, i+k]$. However, this might not always be necessary. If σ is sufficiently small and k is sufficiently large then I_1 may stand no chance of obtaining $\varphi(I_1) > t$. For any x, setting $I_x = [i, \ldots, i+k+x]$ (skipping the next $(x-1)$ intervals), an upper bound on the score of I_x is given by $\varphi(I_x) \leq \frac{\sigma+mx}{\sqrt{k+x}}$. Thus, $\varphi(I_x)$ has a chance of surpassing t only if $t \leq \frac{\sigma+mx}{\sqrt{k+x}}$. Solving for x, we obtain $x \geq (t^2 - 2m\sigma + \sqrt{4k - 4m\sigma + t^2})/2m^2$. Thus, the next ending point to consider is $i + h - 1$, where $h = \lceil k + x \rceil$.

The improvement in efficiency depends on the number of ending points that are skipped. This number, in turn, depends on the tightness of the two bounds t and m. Initially, t may be set to $t = \max_i w_i$. As we proceed, t is replaced by the maximal score encountered so far, gradually improving performance.

Parameter m provides an upper bound on the values of single elements in W. In the description above, we used the *global* maximum $m = \max_i w_i$. However, using this bound may limit the usefulness of the LookAhead approach, since even a single high value in the data will severely limit the skip size. Thus, we use a *local* approach, where we bound the value of the single elements within a window of size κ. The most efficient definition of local bounds is the *slope maximum*: For a window size κ, for all $1 \leq i < n$, precalculate, $f_i = \max_{i<j\leq i+\kappa} \frac{\sum_{\ell=i+1}^{j} w_\ell}{j-i}$. Although f_i is not an upper bound on the value of elements within the κ-window following i, the value of $f_i x$ is indeed an upper bound on $\sum_{j=i+1}^{i+x} w_j$ within the κ-window, maintaining the correctness of the approach. Note that the skip size must now be limited by the window size: $h = \min(\lceil k + x \rceil, \kappa)$. Thus, larger values of κ give better performance although preprocessing limits the practical window size to $O(\sqrt{n})$.

The psuedocode in Algorithm 2 summarizes this variant of the LookAhead algorithm.

Algorithm 2. LookAhead Algorithm—Fixed Sample Case

Input: Sequence W, window size κ.
Output: The maximum-scoring interval I.

Preprocessing
$t = \max_i w_i$, $I = [\text{argmax}_i w_i]$
$sum[1, 0] = 0$;
for $i = 1$ to n **do**
  $f_i = max_{i<j\leq i+\kappa} \frac{\sum_{\ell=i+1}^{j} w_\ell}{j-i}$
  $sum[1, i] = sum([1, i-1]) + w_i$

$maxScore = 0$
for $i = 1$ to n **do**
  $\sigma = w_i$, $k = 1$
  while $i + k - 1 < n$ **do**
    $x = \left\lceil (t^2 - 2m\sigma + \sqrt{4k - 4m\sigma + t^2})/2m^2 \right\rceil$ (*)
    $k = k + \min(\kappa, x)$
    $\sigma = sum([1, i+k]) - sum([1, i-1])$
    $score = \frac{\sigma}{\sqrt{k}}$
    if $score > maxScore$ **then**
      $maxScore = score$, $I = [i, i+k-1]$

A simple variant of the LookAhead algorithm allows us to limit the search for the maximal scoring interval to intervals starting within a fixed zone Z_1 and ending with another fixed zone Z_2. This is obtained by limiting the two loops in the algorithm to the required ranges. The GFA algorithm, described shortly, uses this variant of the algorithm.

Class discovery. Application of the LookAhead heuristic to the class discovery mode is more complex, since the size of the subset C optimizing the score of each interval may vary. The following variation accommodates this difficulty, albeit at reduced algorithmic efficiency. Given the matrix W over a samples set S, create a vector $d = d_i$ as follows. For each $1 \leq i \leq n$, order the set $\{w_{s,i} : s \in S\}$ in decreasing order: $w_{s_1,i} \geq \ldots \geq w_{s_{|S|},i}$. Set $d_i = \max_{j:1\leq j\leq |S|} \frac{\sum_{\ell=1}^{j} w_{s_\ell,i}}{\sqrt{j}}$. The vector d is used only in the preprocessing

step, to compute the slope values f_i. At the main loop of the algorithm, the score of a specific interval $[i, i+k-1]$ is computed from the original data matrix W.

The correctness of the variant follows from the observation that if the subset C_1 maximizes the score of interval I_1 and subset C_2 maximizes the score of an extended interval I_2 (i.e. $I_1 \subset I_2$), setting $x = |I_2| - |I_1|$, then

$$opt(I_2) = \frac{\sum_{s\in C_2}\sum_{i\in I_2} w_{s,i}}{\sqrt{|I_2|\cdot|C_2|}} \le \frac{\sum_{s\in C_2}\sum_{i\in I_1} w_{s,i} + f_i x\sqrt{|C_2|}}{\sqrt{|I_2|\cdot|C_2|}} \tag{15}$$

$$\le \frac{\sum_{s\in C_1}\sum_{i\in I_1} w_{s,i}}{\sqrt{|I_1|\cdot|C_1|}} + \frac{f_i x\sqrt{|C_2|}}{\sqrt{|I_2|\cdot|C_2|}} = opt(I_1) + \frac{f_i x}{\sqrt{|I_2|}} = opt(I_1) + \frac{f_i x}{\sqrt{|I_1|+x}}$$

which can then be solved for x in the step marked by $(*)$ in Algorithm 2.

4.2. Geometric family algorithm (GFA)

GFA is based on a combination of the approximation algorithm of Section 3, which is used to zero-in on "candidate zones," and the *LookAhead* algorithm, which is then used to search within these candidate zones.

Specifically, let M be the maximum score of the intervals in $\mathcal{I}$, and let M^* be the maximal score of all intervals. Consider an interval $J \in \mathcal{I}$ with $\varphi(J) < M/\alpha \le M^*/\alpha$. By Lemma 2, if I^* is the optimal interval, then J cannot be the left-most largest interval of $\mathcal{I}$ contained in I^*. Thus, when searching for the optimal interval, we need not consider any interval for which J is the left-most largest interval. For each interval J, we define the *cover zone* of J to be those intervals I for which J is the left-most largest interval. Specifically, for $J = [x, y]$ such that $|J| = k_j$, let L-COV$(J) = x - \Delta_j + 1$ and R-COV$(J) = x + k_{j+1} - 2$. The *cover zone* of J is COVER$(J) = \{I = [u, v] : u \in [\text{L-COV}(J), x], v \in [y, \text{R-COV}(J)]\}$. In GFA, we concentrate only on intervals J with $\varphi(J) \ge M/\alpha$ and search COVER(J) for the optimal interval using the *LookAhead* algorithm. If several intervals overlap, then we combine their cover zones, as described in Algorithm 3

Algorithm 3. Finding the Optimal Interval—Fixed Sample Case

Input: $W = \{w_i\}_{i=1}^n$.
Output: The maximum-scoring interval $I_{\max}$.

forall $J \in \mathcal{I}$ compute $\varphi(J)$
$M = \max_{J\in\mathcal{I}}\{\varphi(J)\}$, $U = \{J \in \mathcal{I} : \varphi(J) > M/\alpha\}$
while $U \neq \emptyset$ **do**
  $J' = \text{argmax}_{J\in U}\{\varphi(J)\}$, $U' = \{J \in U : J \cap J' \neq \emptyset\}$
  L-COV$(U') = \min\{\text{L-COV}(J) : J \in U'\}$ and R-COV$(U') = \max\{\text{R-COV}(J) : J \in U'\}$
  $J_\cap = \cap_{J\in U'} J$. Denote $J_\cap = [\text{l-}J_\cap, \text{r-}J_\cap]$
  Run *LookAhead* to find the max score interval among the intervals starting in [L-COV(U'), l-$J_\cap$] and ending in [r-$J_\cap$, R-COV(U')]. Denote the optimal by $I_{opt}(U')$.
  $U = U - U'$
$I_{\max} = \text{argmax}_{I_{opt}(U')}\{\varphi(I_{opt}(U'))\}$
return $I_{\max}$

Class discovery. The algorithm for the class discovery case is identical to that of the fixed sample case except that instead of using $\varphi(J)$ we use $opt(J)$, and using Algorithm 4.

4.3. Finding multiple aberrations

In the previous section, we showed how to find the aberration with the highest score. In many cases, we want to find the k most significant aberrations for some fixed k, or to find all aberrations with score beyond some threshold t.

Fixed sample. For the fixed sample case, finding multiple aberrations is obtained by first finding the top-scoring aberration and then recursing on the remaining left and right intervals. We may also find *nested* aberrations by recursing within the interval of aberration. We note that when using the GFA algorithm,

in the recursion we need not recompute the scores of intervals in the geometric family $\mathcal{I}$. Rather, we compute these scores only once. Then, in each recursive step, we find the max score within each region and rerun the internal search (using *LookAhead*) within the candidate cover zones.

Class discovery. For the class discovery case, we must first clearly define the desirable output, specifically, how to handle overlaps between the high scoring rectangles ($I \times C$). If no overlaps are permitted, then a simple greedy procedure may be used to output all nonoverlapping rectangles, ordered by decreasing score. If overlaps are permitted, then it is necessary to define how much overlap is permitted and of what type. We defer this issue to further research.

5. PERFORMANCE BENCHMARKING AND EXAMPLES

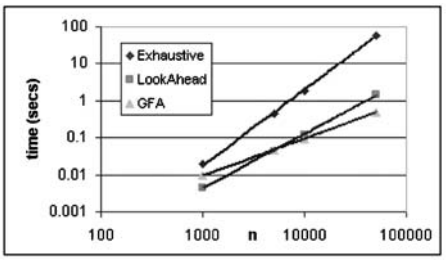
In this section, we present results of applying the scores and methods described in the previous sections to several datasets. In Section 5.1, we benchmark the performance of both the LookAhead and GFA algorithms on synthetic datasets and on data from breast cancer cell-lines (Hyman *et al.*, 2002) and breast tumor data (Pollack *et al.*, 2002). In Section 5.2, we demonstrate results of applying the multiple-sample methods to DCN data from breast cancer samples, using both the fixed sample mode and the class discovery mode.

5.1. Single samples

We benchmark the performance of the exhaustive, LookAhead, and GFA approaches on synthetic data, generated as follows. A vector $V = \{v_i\}$ is created by independently drawing n values from the Normal(0,1) distribution. A synthetic amplification is "planted" in a randomly placed interval I. The length of I is drawn randomly from the distribution Geom(0.01)+10, and the amplitude of the amplification is drawn uniformly in the range $[0.5, 10]$. Synthetic data was created for four different values of n: 1,000, 5,000, 10,000, and 50,000. In addition, we applied the algorithms to six different biological DCN vectors from Pollack *et al.* (2002) and Hyman *et al.* (2002). Benchmarking results of finding the maximum scoring interval are summarized in Table 1. Note that the data from different chromosomes were concatenated for each biological sample to produce significantly long benchmark instances.

TABLE 1. BENCHMARKING RESULTS OF THE EXHAUSTIVE, LOOKAHEAD, AND GFA ALGORITHMS ON SYNTHETIC VECTORS OF VARYING LENGTHS (RAND1-RAND4), AND ON SIX DIFFERENT BIOLOGICAL DCN VECTORS FROM POLLACK *et al.* (2002) (POL) AND HYMAN *et al.* (2002) (HYM)[a]

	Rand1	*Rand2*	*Rand3*	*Rand4*	*Pol*	*Hym*
n	1,000	5,000	10,000	50,000	6,095	11,994
# of instances	1,000	200	100	100	3	3
Exhaustive	0.0190	0.467	1.877	57.924	0.688	2.687
LookAhead	0.0045	0.044	0.120	1.450	0.053	0.143
GFA	0.0098	0.047	0.093	0.495	0.079	0.125



[a]Running times are reported in seconds; simulations performed on a 0.8 GHz Pentium III PC. For Hym and Pol, data from all chromosomes was concatenated to produce significant sized benchmarks. LookAhead was run with $\kappa = \sqrt{n}$, and GFA with $\epsilon = 0.1$. Linear regression to log–log plots suggest that running times of the Exhaustive, LookAhead, and GFA algorithms are $O(n^2)$, $O(n^{1.5})$ and $O(n)$, respectively.

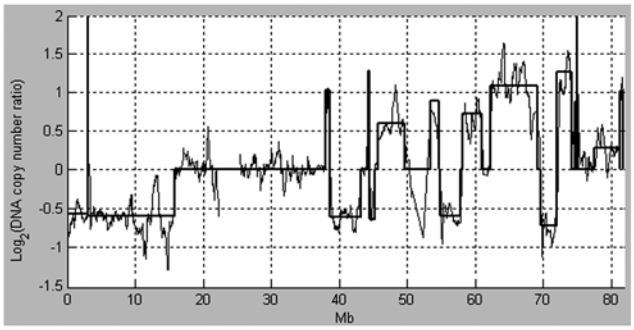


FIG. 1. Significant alterations in breast cancer cell line sample MDA-MB-453, chromosome 17 (7,723 probes). The thin line indicates the raw data, smoothed with a 1Mb moving average window. Overlaid thick lined step function denotes the identified aberrations, where intervals above the x-axis denote amplifications with score $\varphi_S^{sig}(I) > 4$ and intervals below the x-axis denote deletions with score $\varphi_S^{sig}(I) < -4$. The relative y-position of each interval indicates the average signal in the altered interval.

Figure 1 depicts high-scoring intervals identified in chromosome 17, based on aCGH data from a breast cancer cell line sample (MDA-MB-453, 7,723 probes). Here, all intervals with scores > 4 were identified, including nested intervals. Running time of the full search dropped from 1.1 sec per sample with the exhaustive search, to 0.22 sec using the recursive LookAhead heuristic. The running times of the exhaustive, LookAhead, and GFA algorithms exhibit growth rates of $O(n^2)$, $O(n^{1.5})$, and $O(n)$, respectively.

5.2. Multiple samples

Fixed sample set. We searched for common alterations in the two different breast cancer datasets. We used φ^{rob} to detect common aberrations in a set of 14 breast cancer cell line samples (data from Hyman *et al.* [2002]). Figure 2 depicts the chromosomal map of common aberrations that were identified with a score of $|\varphi_S^{rob}(I)| > 4.5$. No intervals with scores of this magnitude were located when the analysis was repeated on random data.

Class discovery. We used $score^{rob}$ to detect classes of common alterations in 41 breast tumor samples (data from Pollack *et al.* [2002]). A large number of significant common alterations were identified in

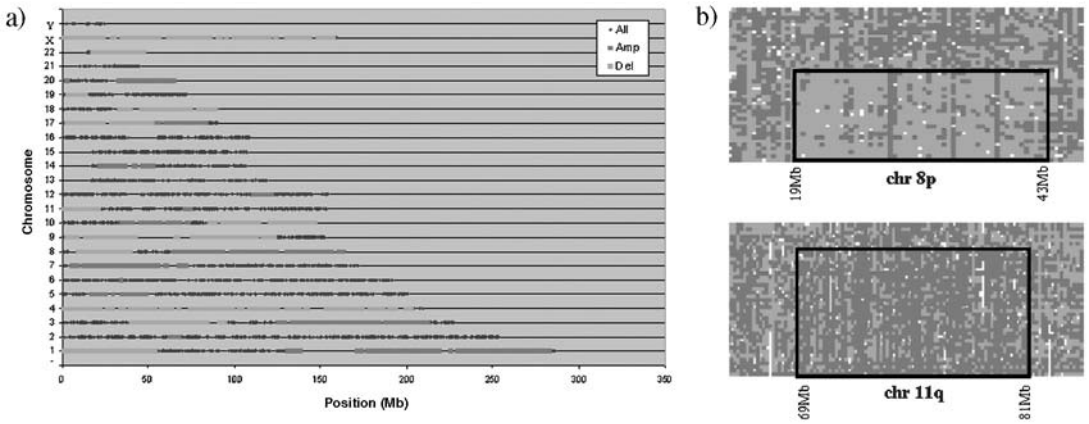


FIG. 2. (**a**) Common alterations detected in 14 breast cancer cell line samples (data from Hyman *et al.* [2002]), in *fixed set* mode. All alterations with score $|\varphi_S^{rob}(I)| > 4.5$ are depicted, amplifications by red marks and deletions by green marks. Dark blue marks denote all probe positions. (**b**) Common deletion in 8p ($\varphi^{rob}(I, C) = 23.6$) and common amplification in 11q ($\varphi^{rob}(I, C) = 22.8$) identified in two subsets of 41 breast cancer samples in *class discovery* mode, $\tau = 0$ (data from Pollack *et al.* [2002]). X-axis denotes chromosomal position. Samples are arbitrarily ordered in y-axis to preserve a different class structure for each alteration. Red positions denote positive values and green denote negative values.

this dataset, with significantly high intervals scores, $\varphi^{rob}(I, C) > 12$. Again, no intervals with scores of this magnitude were located when the analysis was repeated on randomly permuted data. Some large aberrations were identified, including alterations affecting entire chromosomal arms as described in the literature (Pollack *et al.*, 2002). Specifically, amplifications in 1q, 8q, 17q, and 20q and deletions in 1p, 3p, 8p, and 13q were identified, as well as numerous additional smaller alterations. Some of these regions contain known oncogenes (e.g., MYC (8q24), ERBB2 (17q12), CCND1 (11q13), and ZNF217 (20q13)) and tumor suppressor genes (e.g., RB (13q14), TP53 (17p13), BRCA1 (17q21), and BRCA2 (13q13)). Figure 2 depicts two significant common alterations that were identified in 8p (deletion) and 11q (amplification). An interesting aspect of the problem, which we did not attempt to address here, is the separation and visualization of different located aberrations, many of which contain significant intersections.

APPENDIX A. PROOF OF THEOREM 1

Let $f : \mathcal{R} \to \mathcal{R}$ be a strictly concave function (that is, $tf(x) + (1-t)f(y) < f(tx + (1-t)y)$ for all $0 < t < 1$). We define the score of an interval I with respect to f, denoted $S_f(I)$, to be the sum of I's entries divided by $f(|I|)$. In particular, for $f(x) = \sqrt{x}$, $S(I)$ coincides with the interval score function we use throughout this paper.

For a given vector of real numbers of length n, let $m > 0$ denote the maximal interval score obtained (among all $\binom{n}{2}$ intervals). We call an interval *optimal* if its score equals m. The following theorem bounds the number of optimal intervals:

Theorem 4. *There can be at most n optimal intervals.*

Claim. For all $x, y > z > 0$, the following inequality holds: $f(x) + f(y) > f(z) + f(x + y - z)$.

Proof. Define $t = (x - z)/(x + y - 2z)$. Note that

$$x = t(x + y - z) + (1 - t)z, \quad \text{and} \quad y = tz + (1 - t)(x + y - z).$$

As f is strictly concave and $0 < t < 1$,

$$tf(x + y - z) + (1 - t)f(z) < f(t(x + y - z) + (1 - t)(z)) = f(x), \quad \text{and}$$

$$tf(z) + (1 - t)f(x + y - z) < f(t(z) + (1 - t)(x + y - z)) = f(y)$$

Summing the above two inequalities, we get

$$f(x + y + z) + f(z) < f(x) + f(y). \qquad \blacksquare$$

Claim A. If two optimal intervals I and J intersect each other, then either I properly contains J or vice versa.

Claim. If two optimal intervals I and J are disjoint, there must be at least one point between them which is not included in either. Otherwise, consider the interval $K = I; J$ representing their union.

Let $\mathcal{I}$ be a collection of optimal intervals. For each $x \in \{1, \ldots, n\}$, we denote by $\mathcal{I}(x)$ the set of optimal intervals that contains x.

Claim. All intervals in $\mathcal{I}(x)$ have different sizes.

Proof. Any two optimal intervals that contains x intersect. By Claim A, one of them properly contains the other. Thus, they have different size. $\blacksquare$

We define for each x the smallest interval that contains x, denoted by $T(x)$. Note that by the previous claim, there can be only one minimal interval that contains x. To complete the upper bound proof, we show now that the mapping $T : \{1, \ldots, n\} \leftarrow \mathcal{I}$ is onto.

Lemma 3. *For each optimal interval I, there exists a point x such that $T(x) = I$.*

Proof. Let I be an optimal interval, $I = [i, \leftarrow, j]$. We consider two cases:

Case 1. The left-most point i, of I, is not included in any interval $J \subset I$. By Claim A, all the intervals that contains i must contain I, and thus I is the smallest interval that contains i. Therefore $T(i) = I$.

Case 2. Otherwise, let J denote the largest interval that is properly contained in I and $i \in J$. Let k denote the right-most point in J. We show now that $T(k+1) = I$.

First note that since J is properly contained in I and J contains the left-most point of I, J cannot contain the right-most point of I. Therefore $k+1 \in I$. Assume, towards contradiction, that there exists a shorter interval I' that contains $k+1$. As I' intersect I and it is shorter, it must be (Claim A) properly contained in I. Let $i' \geq i$ denote the left-most point of I'. We consider three cases:

- $i' = i$. In this case, $i \in I'$, and $J \subset I'$, a contradiction to the definition of J.
- $i < i' \leq k$. In this, we contradict Claim A as

$$k \in J \cap I', i \in J \setminus I', k+1 \in I' \setminus J.$$

- $i' = k+1$. In this case, $k+1$ is the left-most point of I', while k is the left-most point of J, a contradiction to Claim A.

Therefore, there I is the smallest interval containing $k+1$, and thus $T(k+1) = I$. ■

We conclude with the following theorem.

Theorem 5. *There can be at most n optimal intervals.*

This bound is, in fact, tight as can be seen by considering any prefix of $v_1 = 1, \ldots v_i = f(i) - f(i-1), \ldots$.

APPENDIX B. ALGORITHM FOR COMPUTING *opt*(*I*)

For a matrix W and an interval I, the following algorithm efficiently computes $opt(I)$. Note that we assume that for each s and j, $sum_s([1, j])$ is already known. This can be done in a preprocessing phase, for all s and j in $O(n|S|)$.

Algorithm 4. Computing $opt(I)$

Input: Matrix $W = \{w_{s,i}\}$, Interval $I = [x, y]$.
Output: $opt(I)$.

Foreach $s \in S$ compute $sum_s(I) = sum_s([1, y]) - sum_s([1, x-1])$
Sort the values of $sum_s(I)$ in decreasing order. Let $s_1, \ldots, s_m$ be the order of the s's.
$sum(0) = 0$
for $\ell = 1$ to m **do**
  $sum(\ell) = sum(\ell - 1) + sum(I)_{s_\ell}$
Output $opt(I) = \max_\ell \{sum(\ell)/\sqrt{\ell}\}$

REFERENCES

Balsara, B.R., and Testa, J.R. 2002. Chromosomal imbalances in human lung cancer. *Oncogene* 21(45), 6877–6883.

Bignell, G.R., Huang, J., Greshock, J., Watt, S., Butler, A., West, S., Grigorova, M., Jones, K.W., Wei, W., Stratton, M.R., Futreal, P.A., Weber, B., Shapero, M.H., and Wooster, R. 2004. High-resolution analysis of DNA copy number using oligonucleotide microarrays. *Genome Res.* 14(2), 287–295.

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(14), 4744–4748.

DeGroot, M.H. 1989. *Probability and Statistics*, chap. 5.7, 275, Addison-Wesley, Reading, MA.

Feller, W. 1970. *An Introduction to Probability Theory and Its Applications*, vol. I, chap. VII.6, 193, John Wiley, New York.

Fu, K.S., and Mui, J.K. 1981. A survey of image segmentation. *Pattern Recognition* 13(1), 3–16.

Hedenfalk, I., Ringner, M., Ben-Dor, A., Yakhini, Z., Chen, Y., Chebil, G., Ach, R., Loman, N., Olsson, H., Meltzer, P., Borg, A.A., and Trent, J. 2003. Molecular classification of familial non-BRCA1/BRCA2 breast cancer. *PNAS* 100(5), 2532–2537.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J., and Toivonen, H.T.T. 2001. Time series segmentation for context recognition in mobile devices. *Proc. 2001 IEEE Int. Conf. on Data Mining (ICDM 2001)*, 203–210.

Hupe, P., Stransky, N., Thiery, J.P., Radvanyi, F., and Barillot, E. 2004. Analysis of array CGH data: From signal ratio to gain and loss of DNA regions. *Bioinformatics*.

Hyman, E., Kauraniemi, P., Hautaniemi, S., Wolf, M., Mousses, S., Rozenblum, E., Ringner, M., Sauter, G., Monni, O., Elkahloun, A., Kallioniemi, O.P., and Kallioniemi, A. 2002. Impact of DNA amplification on gene expression patterns in breast cancer. *Cancer Res.* 62, 6240–6245.

Kallioniemi, O.P., Kallioniemi, A., Sudar, D., Rutovitz, D., Gray, J.W., Waldman, F., and Pinkel, D. 1993. Comparative genomic hybridization: A rapid new method for detecting and mapping DNA amplification in tumors. *Semin. Cancer Biol.* 4(1), 41–46.

Lipson, D., Ben-Dor, A., Dehan, E., and Yakhini, Z. 2004. Joint analysis of DNA copy numbers and expression levels. *4th Workshop on Algorithms in Bioinformatics (WABI 04)*.

Mertens, F., Johansson, B., Hoglund, M., and Mitelman, F. 1997. Chromosomal imbalance maps of malignant solid tumors: A cytogenetic survey of 3185 neoplasms. *Cancer Res.* 57(13), 2765–2780.

Morley, M., Molony, C.M., Weber, T.M., Devlin, J.L., Ewens, K.G., Spielman, R.S., and Cheung, V.G. 2004. Genetic analysis of genome-wide variation in human gene expression. *Nature* 430(7001), 743–747.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W.L., Chen, C., Zhai, Y., Dairkee, S.H., Ljung, B.M., Gray, J.W., and Albertson, D.G. 1998. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genet.* 20(2), 207–211.

Platzer, P., Upender, M.B., Wilson, K., Willis, J., Lutterbaugh, J., Nosrati, A., Willson, J.K., Mack, D., Ried, T., and Markowitz, S. 2002. Silence of chromosomal amplifications in colon cancer. *Cancer Res.* 62(4), 1134–1138.

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. *Nature Genet.* 23(1), 41–46.

Pollack, J.R., Sorlie, T., Perou, C.M., Rees, C.A., Jeffrey, S.S., Lonning, P.E., Tibshirani, R., Botstein, D., Borresen-Dale, A., and Brown, P.O. 2002. Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *PNAS* 99(20), 12963–12968.

Polzehl, J., and Spokoiny, S. 2000. Adaptive weights smoothing with applications to image restoration. *J. Royal Statist. Soc. B* 62, 335–354.

Sebat, J., Lakshmi, B., Troge, J., Alexander, J., Young, J., Lundin, P., Maner, S., Massa, H., Walker, M., Chi, M., Navin, N., Lucito, R., Healy, J., Hicks, J., Ye, K., Reiner, A., Gilliam, T.C., Trask, B., Patterson, N., Zetterberg, A., and Wigler, M. 2004. Large-scale copy number polymorphism in the human genome. *Science* 305(5683), 525–528.

Address correspondence to:
Doron Lipson
Computer Science Dept.
Technion
Haifa, Israel 32000

E-mail: dlipson@cs.technion.ac.il