Pacific Symposium on Biocomputing 15:371-382(2010)

ÇOKGEN: A SOFTWARE FOR THE IDENTIFICATION OF RARE COPY NUMBER VARIATION FROM SNP MICROARRAYS

GÖKHAN YAVAŞ[1], MEHMET KOYUTÜRK[1,3], MERAL ÖZSOYOĞLU[1], MEETHA P. GOULD[2], THOMAS LAFRAMBOISE[2,3]

[1]*Department of Electrical Engineering & Computer Science,* [2]*Department of Genetics,* [3]*Center for Proteomics & Bioinformatics*

Case Western Reserve University, Cleveland, OH, 44106, USA

Until fairly recently, it was believed that essentially all human cells harbor two copies of each locus in the autosomal genome. However, studies have now shown that there are segments of the genome that are polymorphic with regard to genomic copy number. These copy number variations (CNVs) have a role in various diseases such as Alzheimer disease, Crohn's disease, autism and schizophrenia. In the effort to scan the entire genome for these gains and losses of DNA, single nucleotide polymorphism (SNP) arrays have emerged as an important tool. As such, CNV identification from SNP array data is attracting considerable attention as an algorithmic problem, and many methods have been published over the last few years. However, many of the existing model-based methods train their models based on common variations and are therefore less successful in the identification of rare CNVs, detection of which may be very important in personalized genomics applications. In this paper, we formulate CNV identification explicitly as an optimization problem with an objective function that is characterized by several adjustable parameters. These parameters can be configured based on the characteristics of the experimental platform and target application, so that the solution to the optimization problem is the most accurate set of CNV calls. Our method, termed ÇOKGEN, efficiently solves this problem using a variant of the well-known heuristic simulated annealing. We apply ÇOKGEN to data from hundreds of samples, and demonstrate its ability to detect known CNVs at a high level of sensitivity without sacrificing specificity, not only for common but also rare CNVs. Furthermore, we show that it performs better than other publicly-available methods. The configurability of ÇOKGEN, its computational efficiency, and its accuracy in calling rare CNVs make it particularly useful for personalized genomics applications. ÇOKGEN is implemented as an R package and is freely available at http://mendel.gene.cwru.edu/laframboiselab/software.php.

1. Introduction

Identification of DNA variants that contribute to disease is a central aim in human genetics research and has immediate applications in personalized genomics. Pinpointing these causal loci requires the ability to accurately assess DNA sequence variation, on a genome-wide scale. In recent years, considerable progress has been made in identifying and cataloging single-nucleotide polymorphisms (SNPs) in many populations [1]. Commercial SNP microarray platforms can now genotype, with >99% accuracy, over one million SNPs in an individual in one assay [2, 3].

The discovery of copy number variants (CNVs) as a significant source of variation has complicated the identification of genetic differences among humans. CNVs are defined as chromosomal segments, at least 1000 bases (1 kb) in length that vary in number of copies from human to human [4-8]. Since their discovery, several high-profile studies have been published associating copy number variation in the genome with a variety of common diseases. Recent examples include Alzheimer disease [9], Crohn's disease [10], autism [11], and schizophrenia [12]. The significance of the gains (copy number greater than two) and losses (copy number less than two) that comprise these variants is increasingly evident, and cataloging them and assessing their frequencies has become an important goal.

SNP arrays contain hundreds of thousands of unique nucleotide probe sequences, each designed to hybridize to a target DNA sequence. When a DNA sample is properly prepared and applied to the array, specialized equipment can produce a measure of the intensity of hybridization between each probe and its target in the sample. The

underlying principle is that the hybridization intensity depends upon the amount of target DNA in the sample, as well as the affinity between target and probe. Extensive processing and analysis of these raw intensity measures yield estimates of some characteristic of the target sequences in the sample - either target quantity [13, 14], base composition [15, 16], or both. In copy number inference, the objective is to identify chromosomal regions at which the number of copies per cell deviates from two. These include gains and losses.

There is now a large body of literature describing algorithms to infer copy number from SNP array data. All such algorithms address one or more of the three general steps: normalization, raw copy extraction, and CNV calling. Normalization is performed on the raw array intensity data in order to be able to compare these values fairly, thereby taking into account differences in overall array brightness and additional sources of nuisance variation. Raw copy number extraction entails converting the multiple measurements for each genomic site into a single raw measure of copy number. The word "raw" here indicates that measurements from surrounding loci are not yet taken into account, and the measure is permitted to be non-integer. However, since gains and losses occur in discrete segments often encompassing several such loci, true copy number is locally constant. Consequently, the final CNV calling step takes advantage of this fact, smoothing or segmenting the raw copy numbers into discrete segments of consistent copy number.

For the Affymetrix platform, the community has largely settled upon quantile normalization [17] as a simple, yet effective, normalization method. The next step, raw copy number extraction, typically entails fitting some model to raw probe intensity data [18-21]. Methods devoted the final step – making CNV calls from raw copy number data – are numerous, and employ various strategies. Three commonly-used strategies are hidden Markov models (HMMs) [21, 22], circular binary segmentation [23, 24], and adapted weight smoothing [25, 26]. Although these methods appear to be quite different from one another in terms of the computational or statistical model they incorporate, at the core of each is an objective function whose optimum solution yields the method's copy number inference for a region. Each objective function is defined by the observed data (raw copy number) and is a function of inferred state (copy number call). The sequence of copy number calls (states) that optimizes the objective function gives the CNV call for each method.

In this paper, we describe a software tool, ÇOKGEN, which implements a novel optimization algorithm for identification of CNVs from raw copy number, based on an objective function that is composed of several explicitly formulated objective criteria. These criteria are carefully designed to quantify the desirability of a CNV assignment with respect to various biological insights and experimental considerations. Our general approach is to first apply a signal processing method to aggressively flag candidate gains and losses. The objective function is then optimized on each region and flanking sequence, yielding final CNV calls and boundaries. Note that the optimization process also filters out many candidate regions; that is, complete rejection of a candidate region is quite possible, as it is part of the solution space for the corresponding optimization problem. This two-step procedure has the advantages of drastically reducing the computational time necessary to find the set of solutions, while identifying precise boundaries for each putative CNV.

A key feature of our method is that it is highly configurable, allowing researchers to define their own objective functions and tune parameters to emphasize relative importance of different objective criteria. We demonstrate with a simple objective function involving a linear combination of variability, parsimony, and length, which performs surprisingly well. We evaluate the performance of our method on Affymetrix 6.0 array data from 270 HapMap individuals [1]. These samples are increasingly well characterized with regard to CNVs and include 60 mother-father-child trios. Therefore, they serve as an excellent benchmark data set. We show via systematic *in silico* studies that it compares favorably with two methods that are currently publicly available. These results demonstrate the proposed method's potential to uncover human genetic variation that other computational approaches may miss.

ÇOKGEN is implemented as an R package that works from the raw binary .CEL files produced by the Affymetrix protocol. It performs the steps including intensity extraction, quantile normalization, raw copy extraction, and CNV extraction (wherein the user may specify the desired objective function). Its graphical tools also allow the user to manually inspect the raw copy number data to gauge confidence in each putative aberration.

2. Methods

ÇOKGEN takes as input the raw .CEL files, and produces a table of inferred gains and losses, genome-wide. It provides a configurable platform for CNV identification, in that it allows users to (i) adjust the parameters of our default formulation to tune the behavior of the method to the target application (e.g., aggressive vs. conservative in calling CNVs), and (ii) specify their own target objective functions. ÇOKGEN also produces "zoomable" plots of raw copy number at the chromosome and sub-chromosome level for manual inspection of identified copy numbers. Details for each step of the framework implemented in ÇOKGEN are described in the following subsections.

2.1. *Intensity Extraction and Normalization of Raw Data*

The raw probe intensities for each array are encoded in the binary .CEL files output by the Affymetrix instrument, one file for each array. As a first step, we use the R package *affxparser* [27] to extract the intensities for each array locus from .CEL files. Next, we quantile normalize [17] the intensities across all arrays in the experiment. This enables fair comparison of intensities, taking into account systematic non-biological differences such as overall array brightness.

2.2. *Raw Copy Number for SNP and CN Markers*

The genomic loci interrogated on the Affymetrix 6.0 array fall into two categories – SNP markers and copy number (CN) markers. The array contains 887,876 autosomal CN and 869,224 autosomal SNP markers, for a total of 1,757,100 (we discard the X and Y chromosomes to avoid gender complications, as well as mitochondrial markers). The markers are ordered from i = 1 to ~1.8 million according to genomic coordinates. A SNP marker is interrogated by either six or eight probes – half for each of the *A* and *B* alleles – and hence produces six or eight normalized intensity measurements for each array. Since the vast majority of SNP markers have six probes, we present that case here. Let A_{i1}, A_{i2}, A_{i3}, B_{i1}, B_{i2}, and B_{i3} denote the three *A* allele and three *B* allele measurements for a SNP marker *i*. Our aim is to produce allele-specific raw copy numbers A_i and B_i for the two alleles such that the distance from the origin in (*A*, *B*) Cartesian coordinates produces a raw measure of the copy number at the i^{th} marker. Toward this end, we linearly rescale the intensities so that $\sqrt{A_i^2 + B_i^2}$ is approximately equal to 2.0, regardless of genotype, for markers that are already deemed to have normal copy numbers (i.e., two copies).

We fit the model

$$Z_i^{(A)} = \alpha_{i1}^{(A)} A_{i1} + \alpha_{i2}^{(A)} A_{i2} + \alpha_{i3}^{(A)} A_{i3} + \beta_{i1}^{(A)} B_{i3} + \beta_{i2}^{(A)} B_{i2} + \beta_{i3}^{(A)} B_{i3} + e_i^{(A)} \quad (1)$$

via least-squares regression, where $Z_i^{(A)}$ is the rescaled (true) copy number for allele A at SNP *i*; $\alpha_{ij}^{(A)}, \beta_{ij}^{(A)}$ for $1 \le j \le 3$ are model parameters, and $e_i^{(A)}$ is the error term. More specifically, in the absence of copy number variation, $Z_i^{(A)}$ is 2.0 for an *AA* genotype, $\sqrt{2}$ for an *AB* genotype, and 0 for a *BB* genotype. The fitting procedure yields estimates $\hat{\alpha}_{i1}^{(A)}, \hat{\alpha}_{i2}^{(A)}, \hat{\alpha}_{i3}^{(A)}, \hat{\beta}_{i1}^{(A)}, \hat{\beta}_{i2}^{(A)}, \hat{\beta}_{i3}^{(A)}$ for the model parameters. We model *B* allele copy number in a similar manner, and obtain estimates $\hat{\alpha}_{i1}^{(B)}, \hat{\alpha}_{i2}^{(B)}, \hat{\alpha}_{i3}^{(B)}, \hat{\beta}_{i1}^{(B)}, \hat{\beta}_{i2}^{(B)}, \hat{\beta}_{i3}^{(B)}$ for the model parameters, quantifying the relationship between *B* allele copy number and the six probe intensities. The objective here is to capture the individual responsiveness of each probe to varying quantities of DNA harboring the *A* and *B* alleles.

Note, however, that fitting the models requires *a priori* knowledge of the genotypes for the values of true allelic copy numbers $Z_i^{(A)}$ and $Z_i^{(B)}$. Affymetrix's default algorithm is quite precise (over 99.5% accurate) for diploid genotyping. Hence, if we were able to avoid samples with duplications and deletions, we could use the genotypes generated by Affymetrix as observed values of *A* and *B* copy numbers. Obviously, we cannot assume knowledge of which samples harbor gains and losses. However, we can utilize basic knowledge on the distribution of copy numbers as evidence suggests that gain and loss events almost always appear in the small minority variant in the population [28]. Therefore, if we define total probe intensity at marker *i* as $PI_i = \sum_{j=1}^{3} A_{ij} + \sum_{j=1}^{3} B_{ij}$, we can safely assume in general that most of the middle two quartiles, across all samples, of PI_i are from individuals with two copies of the chromosomal segment that contains marker *i*. In other words, the individuals that fall into these quartiles for the corresponding marker are likely to carry diploid genotypes *AA*, *AB*, or *BB*. Consequently, we fit

the model based on these samples' genotypes. Note that, in rare cases, it is possible that the dominant allele in the population may deviate from copy number two. In these cases, the proposed method will still detect the CNV but copy number two individuals will appear as having losses or gains at those loci.

Given the 12 parameter estimates for a SNP marker i, we generate raw estimates of A and B copy numbers for all samples by re-applying the model to each sample's six probe intensities. That is, for a sample with probe intensity values A_{i1}, A_{i2}, A_{i3}, B_{i1}, B_{i2}, and B_{i3}, the raw A and B allele copy estimates are A_i and B_i where

$$A_i = \hat{\alpha}_{i1}^{(A)} A_{i1} + \hat{\alpha}_{i2}^{(A)} A_{i2} + \hat{\alpha}_{i3}^{(A)} A_{i3} + \hat{\beta}_{i1}^{(A)} B_{i1} + \hat{\beta}_{i2}^{(A)} B_{i2} + \hat{\beta}_{i3}^{(A)} B_{i3} \quad (2)$$

$$B_i = \hat{\alpha}_{i1}^{(B)} A_{i1} + \hat{\alpha}_{i2}^{(B)} A_{i2} + \hat{\alpha}_{i3}^{(B)} A_{i3} + \hat{\beta}_{i1}^{(B)} B_{i1} + \hat{\beta}_{i2}^{(B)} B_{i2} + \hat{\beta}_{i3}^{(B)} B_{i3} \quad (3)$$

Finally, using these estimates, we calculate the raw copy number R_i at marker i as the distance from the origin in the (A, B)-plane: $R_i = \sqrt{A_i^2 + B_i^2}$.

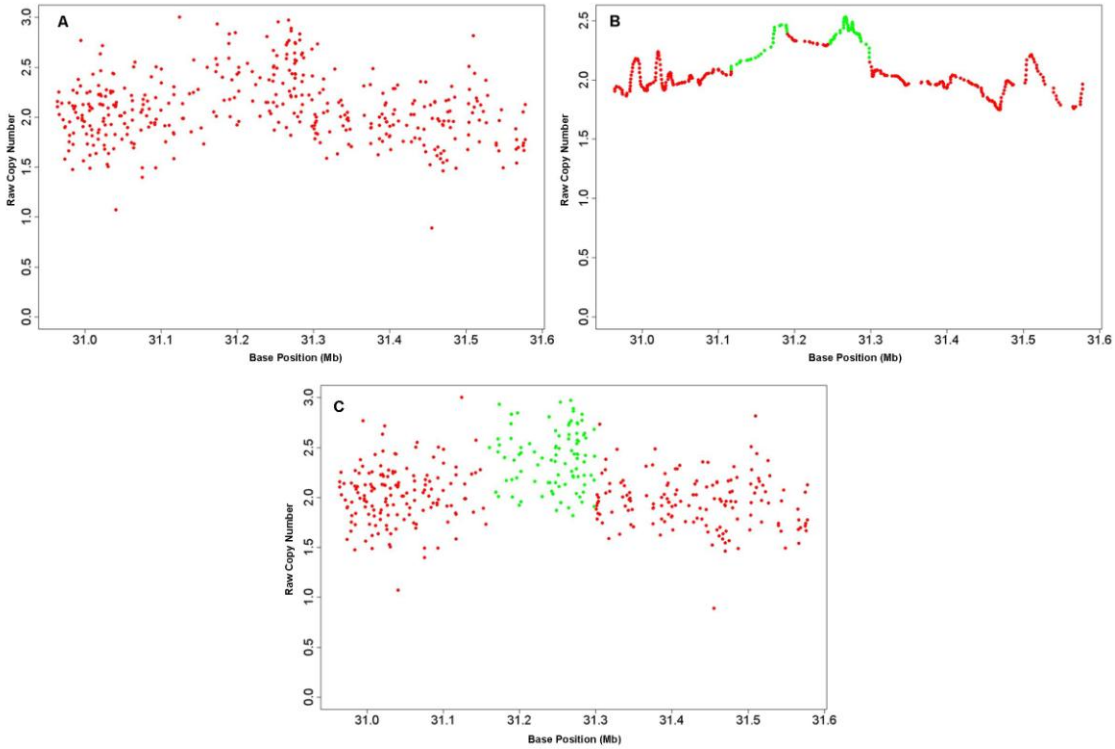

Figure 1. Raw copy numbers for sample NA12763 in a chromosome 12 region. In (A), the raw copy numbers, R_i, for the specified region, are presented. (B) The identified candidate gain regions on the smooth signal R_i^*, which is obtained by applying a low pass filter to R_i. The green colored markers indicate a "gain" class value assignment, whereas the red markers indicate "normal" class assignment by the edge detection algorithm. (C) Optimization of the objective function using simulated annealing makes the final assignments to the markers.

Note that approximately half of the marker loci represented on the 6.0 array correspond to CN markers. Since these markers are each measured by only one probe, they must be treated separately. As above, we consider the samples within the middle two quartiles of (normalized) total probe intensity for the marker to be representative of individuals with copy number two. Therefore, the scaling factor $\hat{\beta}_i$ for CN marker i is the least-squares estimate of the parameter β_i from the model

$$2 = \beta_i PI_i + e_i \quad (4)$$

fit to the middle two quartiles of the normalized probe intensities PI_i. Again, e_i is the error term. The raw copy number for a sample with CN probe intensity PI_i is then calculated as $R_i = \hat{\beta}_i PI_i$.

Using these two separate procedures for SNP and CNV markers yields raw copy numbers R_i for all markers i from 1 to ~1.8 million. Figure 1A, which is generated by ÇOKGEN, gives an example of raw copy numbers for a 394-marker region.

2.3. *Copy Number Variant Detection using Optimization*

Key to our approach is the observation that CNV identification can be formulated explicitly as an optimization problem without any requirement of reference models or training data. Based on general knowledge of the microarray technology and basic biological insights on copy number variation, we specify various quantitative measures that gauge the suitability of copy number assignments based on observed array intensities. We then formulate an objective function that captures the trade-off between these measures, so that the minima of this function represent optimal CNV assignments. This function is characterized by user-defined parameters, allowing the user to tune the performance of algorithms based on the requirements of the specific application (*e.g.*, minimizing false positives due to the cost of experimental verification *vs.* minimizing false negatives to capture existing variation comprehensively).

Formally, the objective of CNV identification is to find a mapping S: {1,..., N} → C, where {1,..., N} denotes the ordered set of markers for the whole genome and C = {C_+, C_0, C_-} is the set of the gain, normal and loss classes, denoted respectively as C_+, C_0 and C_-. Thus, our objective is to assign a class value from C to each marker on genome based on the R_i values such that the class assignment of consecutive markers and their raw copy number estimates are as consistent as possible.

In next subsections, we introduce the objective criteria that are included in the default objective function implemented in ÇOKGEN and the motivation behind these criteria. Researchers may wish to design an objective function of their choice, and indeed our software takes the objective function as an argument precisely to accommodate this. We describe the function as applied to a chromosome with M markers since each chromosome is processed separately.

2.3.1. Variability in raw copy numbers within each copy class should be minimized

The R_i for markers in each gain or loss region should be separable from normal regions. Therefore, CNV identification lends itself to a clustering-like problem – one of partitioning the R_i's into three classes so as to minimize the internal variability of each class. For a given CNV assignment S, we define the set of markers assigned to class c on a chromosome with M markers as $\Pi(c) = \{i \in \{1,\ldots,M\} : S(i) = c\}$ and $\mu_c = \frac{\sum_{k\in\Pi(c)} R_k}{|\Pi(c)|}$ denotes the mean raw copy number for class c. Then, the total intra-class variability induced by this assignment is given by

$$\sigma(S) = \sum_{c\in\{C_+,C_0,C_-\}} \sum_{k\in\Pi(c)} |R_k - \mu_c| \qquad (5)$$

Consequently, a desirable S is expected to minimize $\sigma(S)$ (subject to other constraints). Note that this formulation does not make any assumption about the expected raw copy numbers of the markers and therefore is robust to any systematic bias that might be encountered in measurement and normalization of R_i.

2.3.2. Parsimony principle: Observed variability should be explained via minimum number of anomalies

In general, there are relatively few regions of gain or loss in an individual's genome, relative to normal regions. Therefore, the CNV calls should be as contiguous as possible. Motivated by this observation, we formulate the parsimony principle as a criterion that seeks to minimize the total number of copy number state changes induced by a CNV assignment on the chromosome. Formally, for given CNV assignment S, we define total cut as the number of pairs of adjacent markers that are assigned different copy numbers, $\chi(S) = \sum_{k=1}^{M-1} I(S(k) \neq S(k+1))$. Here I(.) denotes the indicator function (i.e., it is equal to 1 if the statement being evaluated is true, and 0 otherwise).

2.3.3. Filtering out noise by eliminating smaller regions

Longer CNVs indicate higher confidence as it can be statistically argued that shorter sequences of markers with deviant raw copy numbers are more likely to be observed due to noise. Thus, we explicitly consider CNV length as

an additional objective criterion. To do so, we first define a CNV region, *r*, as a maximal set of contiguous markers all assigned to the same copy number state in $\{C_+, C_-\}$, and $\varsigma(S)$ denotes the set of all CNV regions. Furthermore, we denote the number of markers in the CNV region *r* by *l*(*r*). We then define $\lambda(S) = \sum_{r \in \varsigma(S)} \frac{1}{e^{l(r)}}$ as an objective criterion that penalizes shorter CNVs (*e* denotes the natural logarithmic base).

2.3.4. Filtering out noise by eliminating possible false positives

Candidate CNVs with a median raw copy number much larger or much smaller than two indicate higher confidence since a CNV region with median raw copy number close to two is less likely to be valid. For this reason, we require that the median raw copy number of a called loss be below a certain threshold (T_{loss}) and the median for a called gain be above a certain threshold (T_{gain}). We define $\varsigma^+(S)$ and $\varsigma^-(S)$ as the set of all CNV gain and loss regions, induced by assignment *S*, respectively. Furthermore, median(*r*) denotes the median raw copy number value of the markers in the region *r*. We now incorporate $\delta(S) = \sum_{r \in \varsigma^+(S)} I(\text{median}(r) < T_{\text{gain}}) + \sum_{r \in \varsigma^-(S)} I(\text{median}(r) > T_{\text{loss}})$ into the objective function to minimize the effects of the noisy signal. Here, T_{gain} and T_{loss} are user-defined parameters which basically define the upper and lower limits for the raw copy number of markers in the set $\Pi(C_0)$ (i.e., the set of markers assigned to the normal class). As T_{gain} is increased and T_{loss} is decreased, candidate regions are penalized more harshly. In our experiments, we use 2.35 and 1.65 for T_{gain} and T_{loss} respectively, since these values provide reasonable performance.

2.3.5. Putting the pieces together: A single objective function for CNV identification

We use a linear combination of the criteria above as an objective function. Namely, we define the optimal copy number assignment as the mapping S^*: $\{1,\ldots, N\} \rightarrow C = \{C_+, C_0, C_-\}$ such that the function

$$f(S) = k_\sigma \sigma(S) + k_\chi \chi(S) + k_\lambda \lambda(S) + k_\delta \delta(S) \qquad (6)$$

is minimized at $S = S^*$. We briefly talk about how these parameters are adjusted in section 3.5.

2.4. *Two Phase CNV Identification*

Since the solution space of the optimal copy number assignment problem is exponential in the number of markers we require a good initial solution and a heuristic algorithm which iteratively improves the solution. For this purpose, we use a two-phase algorithm: (i) we first determine a set of candidate gain and deletion regions via a filtering and aggressive edge detection procedure which we consider as an initial CNV assignment, $S^{(0)}$; (ii) we employ an iterative improvement based algorithm to adjust the boundaries of duplications and deletions accurately, and eliminate false positives.

In order to identify the boundaries for CNV regions, it is necessary to smooth the raw copy number signal since it is highly noisy. We use a simple discrete low-pass filter with filter kernel [1/3; 1/3; 1/3], i.e., the first filtered copy number estimate is given by $R_i^{(1)} = \frac{R_{i-1} + R_i + R_{i+1}}{3}$. Applying the filter for a second time, we obtain $R_i^{(2)} = \frac{R_{i-1}^{(1)} + R_i^{(1)} + R_{i+1}^{(1)}}{3} = \frac{R_{i-2} + 2R_{i-1} + 3R_i + 2R_{i+1} + R_{i+2}}{9}$. Consequently, introducing an adjustable repetition parameter *W*, we obtain $R_i^* = R_i^{(W)}$ as a smooth version of the copy number intensity for a user defined value of *W*. Here, larger *W* provides smoother signals, thereby eliminating false positives, at the cost of missing true CNVs that span a smaller number of markers. For the ÇOKGEN's default value, we chose *W*=20, for which we obtain reasonable results. Figure 1B demonstrates how the raw copy numbers R_i in Figure 1A is converted into a smooth signal R_i^* using the low pass filter.

2.4.1. Identification of Candidate CNV Regions via Edge Detection

Based on the observation that gains and losses manifest themselves as (respectively up or down) concavities in raw copy number of the low-pass filtered data, an edge detection scheme, which we describe below, is a useful tool for the identification of initial CNV assignment $S^{(0)}$. Thus, after low-pass filtering, we apply our edge detection algorithm on the smoothed signal, first identifying high gradient markers that may correspond to transitions between regions with different copy numbers. For this purpose, we interpolate the discrete signal to obtain a real-valued function on the continuous interval $\hat{R}:[0, \mathrm{M}] \to \Re$. This task is performed using the built-in *splinefun* function of R language, which performs cubic spline interpolation of given data points. Next, we generate two sets of high-gradient markers, denoted $D_{\max}$ and $D_{\min}$, for which the function $\hat{R}(i)$ attains maximum increase and maximum decrease, respectively. Specifically, we define

$$\begin{aligned} D_{\max} &= \{i \in 2,\ldots,\mathrm{M} \mid \hat{R}'(i) > \hat{R}'(i-1) \text{ and } \hat{R}'(i) > \hat{R}'(i+1)\} \\ D_{\min} &= \{i \in 2,\ldots,\mathrm{M} \mid \hat{R}'(i) < \hat{R}'(i-1) \text{ and } \hat{R}'(i) < \hat{R}'(i+1)\} \end{aligned} \quad (7)$$

where $\hat{R}'(i)$ denotes the derivative of $\hat{R}(i)$ at marker i. These markers are the approximate inflection points of the signal $\hat{R}(i)$.

Now let Q_{ij} denote the indices corresponding to the set of contiguous markers on the genome starting from marker i and ending at marker j, where $i \le j$. Given the user defined thresholding parameter T_{gain} (see above), we designate Q_{ij} as a candidate gain region (i.e., $\forall k \in Q_{ij}$, $S^{(0)}(k) = C_+$) if it satisfies the following conditions:

1. $i \in D_{\max}$ and $j \in D_{\min}$
2. there exists at least one marker p, $i \le p \le j$, such that $\hat{R}(p) \ge T_{\text{gain}}$
3. $\max(Q_{ij} \cap D_{\max}) < \min(Q_{ij} \cap D_{\min})$
4. Q_{ij} is a maximal set of contiguous markers satisfying the above 3 conditions.

The first condition ensures that the region starts with a marker with locally maximal positive gradient and ends with a marker with locally maximal negative gradient in terms of the raw copy number values. The second condition guarantees that the region contains markers with copy number estimates that might indeed correspond to a gain. The third condition specifies that the region does not contain any interior concavities, i.e. all maximum positive gradient markers in Q_{ij} appear before any maximum negative gradient marker in the region. Finally, condition 4 ensures that Q_{ij} can be enlarged neither at the right nor the left borders. The designation of Q_{ij} as a candidate loss region is done in a completely analogous manner.

All markers m that are not included in a candidate loss or gain region are preliminarily designated as "normal", i.e., $S^{(0)}(m)$ is set to C_0. As a special case, if a candidate gain/loss region identified by edge detection is very close to another candidate region of its type, then we merge these two candidate regions into a single region, since they are likely to correspond to the same aberration.

This procedure gives us an initial CNV identification assignment $S^{(0)}$. As an example, two candidate gain regions identified by the edge detection algorithm are presented in Figure 1B. The green colored markers indicate a "gain" class value assignment, whereas the red markers indicate "normal" class assignment by the described algorithm. Note that, although there are no "loss" class valued markers in the figure, they are colored with blue by ÇOKGEN's visualization tool.

The initial solution is quite aggressive in the sense that many truly normal (copy number two) markers are likely to be placed in the gain or loss classes. To eliminate these false positives and obtain S^*, we use an optimization-based algorithm to tune the boundaries of candidate gain and deletion regions as discussed in the next section.

2.4.2. Fine Tuning of the Region Boundaries using Optimization with Simulated Annealing

This phase of the algorithm begins with initial class assignments, $S^{(0)}$, and iteratively improves it with regard to the value of the objective function f by making moves in a way to quickly reach an optimum and avoid being trapped into undesirable local optima. Note that, while we assume here that $S^{(0)}$ is obtained using the edge detection

procedure presented in the previous section, the optimization procedure presented in this section can be used to refine boundaries generated by any initial segmentation procedure, such as [23, 24].

For a given copy number assignment *S*, we define a *move* as the extension or contraction of a CNV region's boundaries by changing the copy number states assigned to a contiguous group of markers (either inside or outside the region) bordering the region. In short, at each iteration of the algorithm, a random number of contiguous markers is selected from the right or left boundary of a candidate region $Q_{ij} \in \varsigma(S)$ and the corresponding move is defined as the assignment of these markers to either the class of neighboring markers (if the selected markers belong to Q_{ij}) or to Q_{ij}'s class (if the selected markers are outside of Q_{ij}). The concept of a move is illustrated in Figure 2. As seen in the figure, we restrict possible moves to those that can enlarge, shrink, or merge candidate aberrant regions, but can never create a candidate region from scratch or divide a candidate region into two candidate regions. Indeed, we observe that the average distance between two consecutive CNPs reported by McCarroll *et al.* [28] is 2.11 Mb, indicating that it is unlikely for edge detection to misidentify two disjoint CNVs as a single merged CNV.

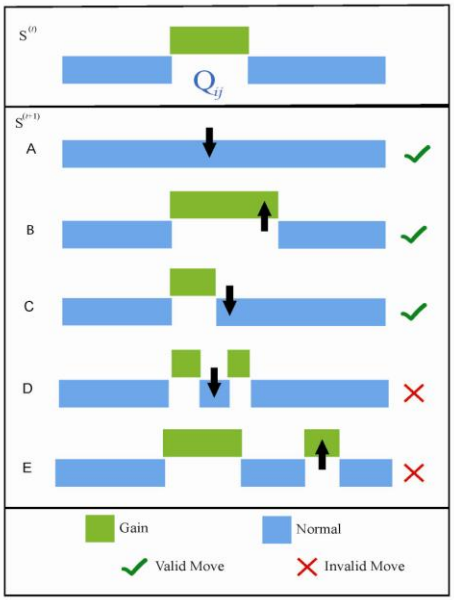

Figure 2. Illustration of the concept of "move" for the proposed iterative improvement algorithm. A valid move is defined as the reassignment of the class of a contiguous group of markers that is within or near a candidate CNV region. At each step of the algorithm, a valid potential move is selected randomly and it is done if it improves the objective function. If it does not improve the objective function, then it is done with probability inversely proportional to its cost on the objective function.

We quantify the quality of a potential move in terms of the difference between the value of the objective function before and after the move, commonly referred to as the *gain* of a move. The gain associated with move *v* is defined as $\gamma(v) = f(S^{(t)}) - f(S^{(t+1)})$ where $S^{(t+1)}$ denotes the copy number assignment if the move *v* were made and $S^{(t)}$ is the current copy number assignment. While it is possible to find the move with maximum gain by exhaustively searching the valid move set, instead we use a stochastic algorithm that is based on simulated annealing [29]. Simulated annealing is an iterative improvement heuristic that proceeds by repeated moves to improve the quality of the solution. Key to its efficiency is the stochastic nature of the selection of moves. At each step, the algorithm first randomly chooses a candidate gain or loss region, Q_{ij}, from the set $\varsigma(S)$ and then chooses a move *v* from the set of all moves that are validly defined on Q_{ij}. If the gain $\gamma(v)$ associated with the candidate move is positive, then the move is made. If the gain is not positive, the move is still made with a certain probability, which is proportional to the gain and declines as a function of time in the course of the algorithm. Therefore, simulated annealing starts its course with aggressive moves to jump out of undesirable local optima, and becomes more conservative as the algorithm proceeds, smoothly converging to a locally optimum solution. In our application, we set an upper limit of five (in terms of number of markers) on the permissible expansion of a CNV region. The procedure is repeated until either there is no positive gain move left to be done on the current solution or a user-defined number of negative gain moves, τ, are already done consecutively, (for our default, we use $\tau = 5$). The mapping obtained at the end of the procedure is reported as *S**.

3. Results and Discussion

We applied our algorithm to Affymetrix 6.0 array data from 270 HapMap individuals. The HapMap samples are divided into African (YRI), Caucasian (CEU) and Asian (CHB/JPT) ethnicities. ÇOKGEN identified a total of 16739 autosomal CNVs over all the samples, for an average of 62 CNVs per individual. Of the 16739 CNVs, 1033 are singletons found uniquely in one individual. A recent study by McCarroll *et al.* [28] identified 1292 autosomal copy number polymorphism (CNP) regions in 270 HapMap samples. Nearly 25% of these CNPs were also

identified by ÇOKGEN The distribution of the CNVs among different ethnicities in the population, as well as the overlap and difference in between the McCarroll *et al.* study and ÇOKGEN are presented in Table 1.

Table 1. The statistics of CNVs identified by ÇOKGEN. The distribution of identified CNVs by ethnicity is shown on the four left-most columns. The comparison of CNPs reported by McCarroll *et al.* and CNVs identified by ÇOKGEN is shown on the two right-most columns. Here McCarroll ∩ ÇOKGEN indicates the counts of CNVs identified by both, whereas McCarroll \ ÇOKGEN indicates the counts identified only by the McCarroll study.

	CEU	YRI	JPT	CHB	Total	McCarroll ∪ ÇOKGEN	McCarroll \ ÇOKGEN
Gains	1711	229	985	786	5777	1357	6145
Losses	3749	370	172	1783	10962	8972	26043
Total	5460	600	271	2569	16739	10329	32188

3.1. *Trio Discordance as a CNV Detection Assessment Tool*

Although CNVs can arise in a *de novo* manner, it is believed that at least 99% of all CNVs in an individual's genome are inherited [28]. The 60 mother-father-child trios in the HapMap data set therefore provide an opportunity to assess the accuracy of CNV detection algorithms by measuring the rate of Mendelian concordance. A CNV in a trio child is said to be Mendelian concordant if it appears in at least one of the parents. Unless the CNV is *de novo*, any discordance is either the result of a false positive call in the child or a false negative call in one of the parents (in rare cases, discordance could also result from a parent harboring a duplication and a deletion at the same locus but on different chromosomal homologs). Discordance rate, while useful, is imperfect as an assessment measure. In particular, it is possible for a CNV identification algorithm to have artificially low discordance rates by calling each CNV in a large number of samples. Even if the samples in which a gain or loss is called are randomly selected, frequently called CNVs will have a lower discordance rate, simply by chance.

3.2. *Performance of ÇOKGEN in Comparison to Existing Software*

We compared the performance of our algorithm with that of two other software packages. The DNA-Chip Analyzer (dChip) [30] is a Windows software package for high-level analysis of gene expression microarrays and SNP microarrays [18, 31]. Birdseye [21] is a rare CNV identification tool based on the hidden Markov models. It is part of the Birdsuite platform [21] ,which is a fully open-source set of tools to detect and report SNP genotypes, common copy number polymorphisms (CNPs), and novel, rare, or *de novo* CNVs in samples processed with the Affymetrix platform.

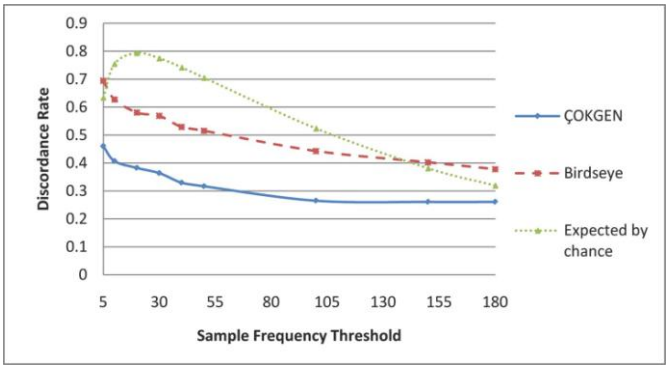

Figure 3. Discordance rate as a function of call frequency strata. For each value of sample frequency threshold, the y-axis shows the average discordance rate for all copy number calls that are less frequent than the threshold.

ÇOKGEN outperformed both Birdseye and dChip in terms of general trio discordance. Overall, it has a 26% discordance rate whereas Birdseye and dChip demonstrate discordance rates of 37% and 93%, respectively on the same array data. dChip was originally optimized for detecting somatic copy number aberrations in cancer cells from earlier versions of the Affymetrix platform. Therefore, Birdseye and ÇOKGEN's superior performance compared to dChip is not surprising. For this reason, we restrict our assessment to ÇOKGEN and Birdseye for the remainder of this section.

As discussed in previous section, the expected discordance rate of an algorithm approaches zero as it calls a CNV in more samples in a data set of trios.

At the extreme, if the algorithm identifies a CNV in all samples, the discordance rate will be zero. Therefore, a more precise assessment of accuracy can be achieved by stratifying discordance rate by call frequency. For this purpose, in Figure 3, we first examine how the discordance rate changes across call frequency strata for ÇOKGEN and Birdseye. As a reference, we also display the expected discordance of randomly called CNVs in this figure. Note that discordance rate is plotted for CNVs with frequencies *at most* the corresponding value on the x-axis. As expected, the performance of both algorithms improves when we consider more frequent CNVs. Nevertheless, it is clear in the Figure 3 that ÇOKGEN outperforms Birdseye significantly at all strata. Furthermore, for up to a frequency threshold of five samples, if the CNV calls were to be done totally at random, it is possible to obtain a better discordance rate than Birdseye. Similarly, for more frequent CNVs, (for frequency threshold larger than 150) random CNV calling performs better than Birdseye at all strata. However, ÇOKGEN performs consistently better than random CNV assignment at all strata which shows its superior performance is not an artifact of the frequency of the CNVs it calls.

Another feature of Figure 3 is Birdseye's sharper decline in discordance rate as the frequency threshold increases. This is likely due to its higher average call frequency as compared to ÇOKGEN. We find that 40% of the concordant CNVs identified by Birdseye have a sample frequency larger than 60, whereas only 20% of the concordant CNVs identified by our algorithm have frequency larger than 60. Concordant CNVs with sample frequency larger than 90 make up 4% of those called by our algorithm as compared to 27% for Birdseye. This clearly shows that ÇOKGEN does not achieve its high concordance rate by overcalling a CNV in multiple samples. When we analyze the density distribution of the discordant CNVs as a function sample frequency for both algorithms, we observe that most of the discordant CNVs for Birdseye are rare whereas more frequent CNVs called by our algorithm turn out to be discordant. These two observations clearly show that ÇOKGEN's performance depends less on the sample frequency and demonstrate its ability to accurately detect rare events.

3.3. *Sensitivity comparison across methods*

Trio discordance is a good hybrid measure of sensitivity (recall) and specificity (precision), but these two measures cannot be easily decoupled based only on discordance rate. A recent study [32] assembled a "stringent dataset" which contains CNVs identified by at least two independent algorithms. The data set contains a total of 808 autosomal CNV regions reported by the study to be harbored in at least one of the 270 HapMap individuals. We use this as a "gold standard" data set in which to evaluate the sensitivity of our method.

ÇOKGEN detects 725 of 808 (≈90%) CNVs from the study presented in [32]. Birdseye obtains the best result by identifying 760 of 808 (≈94%) CNVs. dChip achieves an 89% success rate which is comparable to our method. Therefore, Birdseye seems slightly more sensitive than ÇOKGEN; however, as shown above, this is likely at the cost of a higher false positive rate.

3.4. *Experimental Validation of CNVs not Previously Reported*

To gauge the ability of ÇOKGEN to uncover novel gains and losses, we also compared the CNVs discovered by our method with those in the version 6 (November 2008) of Database of Genomic Variants (DGV) [33]. We used multiplex ligation-dependent probe amplification (MLPA) [34] to verify some of the CNVs which are not reported in the DGV but are identified by ÇOKGEN. The results of these experiments are shown in Table 2. As seen in the table, the copy numbers estimated by MLPA for each of these regions are concordant with the predictions of ÇOKGEN.

Table 2. MLPA results for some of the copy number variantsidentified by ÇOKGEN, which were not previously reported.

Chr	Sample	Bp Start	Bp End	Length (bp)	MLPA Probe Pos.	Type	MLPA
5	NA11830	59753489	59816458	62969	59766589	Gain	2.4
5	NA10846	101261596	101308054	46458	101261461	Loss	1.35
5	NA12144	101256012	101308054	52042	101279312	Loss	1.18
6	NA10846	99225525	99249603	24078	99237564	Loss	1.44

3.5. *Parameter Adjustment*

As explained in section 2.3.5, we have tunable coefficients $k_\sigma, k_\chi, k_\lambda, k_\delta$ that adjust the relative importance of the objective criteria with respect to each other in our objective function. In our experiments, for k_λ and k_δ, we choose large values such as 10^5 and 10^6, respectively, to prohibitively eliminate candidate regions that are likely to be false positive during the course of the algorithm (as opposed to filtering them out in a post-processing phase).

The parameters k_σ and k_χ are used to adjust the apparent trade-off between the "parsimony" and the "variability" components of the objective function. Variability favors the genetic diversity on the genome by permitting many CNVs. On the other hand, according to the parsimony criterion, the variability in the raw copy estimates of markers should be explained via as few CNVs as possible, hence minimizing the number of evolutionary events that have had to occur. Without loss of generality, we require that $k_\sigma + k_\chi = 1$ to highlight the trade-off between these two criteria. To systematically evaluate the effect of these two parameters on performance and determine the best k_σ and k_χ values based on our benchmarking data, we have conducted a series of computational experiments on the sensitivity and trio discordance. Note that lower discordance is desirable, while we want to maximize sensitivity. In our experiments, we have observed that at $k_\sigma = 0.35$ and $k_\chi = 0.65$, trio discordance curve reaches a global minimum and sensitivity starts saturating after a rapid improvement. As k_σ is increased, ÇOKGEN starts behaving less conservatively, which results in a larger number of identified CNVs and improved sensitivity. On the other hand, increased number of CNVs comes with the expense of increased rate of false positives and this manifests itself as a decline in the discordance rate from a certain value of k_σ (in our case, $k_\sigma = 0.35$). Based on these observations, we set $k_\sigma = 0.35$ and $k_\chi = 0.65$ as our defaults.

3.6. *The Software Package*

Our software package, ÇOKGEN, which is implemented in R, processes each sample individually. When a new sample is to be processed, ÇOKGEN first normalizes its probe values to the HapMap distribution, then uses the coefficients obtained from HapMap samples to get raw copy numbers for the new sample. Next, candidate regions are identified using edge detection and marker class assignments are finalized using optimization with simulated annealing. ÇOKGEN is able to output its results in two forms: tabular and graphical. The tabular output is a table of CNV entries with columns: sample ID, chromosome number, CNV start base position, CNV stop base position, and the CNV type. The graphical output allows the user to visualize the results of our CNV identification algorithm. The user can inspect the raw copy signal at any specified part of the genome along with the assigned class values, color-coded (examples are shown in Figure 1). Another aspect of the graphical output is the visualization of the signals of a family together, in which each member represented by a different plotting symbol. This allows the user to see the CNV pattern for the whole family at the same locus of the genome and evaluate the algorithm's trio concordance visually.

Besides its configurability in terms of tuning of parameters, ÇOKGEN also provides the users with the ability to specify their own objective criteria. With this functionality, users can construct their own objective functions that will best suit the characteristics and needs of their own experimental platform and application.

4. Conclusion

We have presented a method to detect germline copy number variants from Affymetrix 6.0 SNP Array data. Our approach, with its accompanying software, will be useful for researchers querying constitutional DNA for association of gains and losses with disease. Indeed, CNVs are emerging as important factors in a growing number of diseases. Although in this paper, we test the ÇOKGEN's performance on only Affymetrix 6.0 SNP array due to its high genome-wide resolution compared to other commercially-available platforms, our software can be easily applied to any platform using different raw copy extraction methods that are suitable for that platform. ÇOKGEN's ability to uncover rare variants is particularly crucial in the context of personalized genomics, as individual-level variation may have a significant impact on human disease. The current work shows that the problem of detecting CNVs from raw array data may be recast as an optimization problem with an explicit objective function. The

objective function chosen here is quite simple and intuitive, but its effectiveness is clear. Our method is wholly contained in a freely-available and flexible software package [35] that efficiently processes raw probe-level .CEL files to produce lists of inferred gains and losses. The software allows the user to tune parameters for the desired specificity-sensitivity balance. With detailed experimental studies on HapMap dataset, we have demonstrated its sensitivity to detect both previously-reported and novel CNVs, while keeping a low false positive rate, as demonstrated by high Mendelian consistency in trios.

Acknowledgments

This work is supported in part by National Science Foundation Award IIS-0916102.

References

1. International HapMap Consortium. *Nature*, **437(7063)**: 1241-2 (2005).
2. Affymetrix. Genome-Wide Human SNP Array 6.0 data sheet. Santa Clara (California) (2007).
3. Illumina. Human1M-duo beadchip data sheet. San Diego (California) (2007).
4. Feuk L, Carson AR, Scherer SW. *Nat Rev Genet*, **7(2)**: 85-97 (2006).
5. Iafrate AJ, Feuk L, Rivera MN, et al. *Nat Genet*, **36(9)**:949-51 (2004).
6. Tuzun E, Sharp AJ, Bailey JA, et al. *Nat Genet*, **37(7)**:727-32 (2005).
7. Redon R, Ishikawa S, Fitch KR, et al. *Nature*, **444(7118)**:444-54 (2006).
8. Korbel JO, Urban AE, Affourtit JP, et al. *Science*, **318(5849)**:420-6 (2007).
9. Rovelet-Lecrux A, Hannequin D, Raux G, et al. *Nat Genet*, **38(1)**:24-6 (2006).
10. Fellermann K, Stange DE, Schaeffeler E, et al. *Am J Hum Genet*, **79(3)**:439-48 (2006).
11. Sebat J, Lakshmi B, Malhotra D, et al. *Science*, **316(5823)**:445-9 (2007).
12. Xu B, Roos JL, Levy S, et al. *Nat Genet*, **40(7)**:880-5 (2008).
13. Zhao X, Li C, Paez JG, et al. *Cancer Res*, **64(9)**:3060-71 (2004).
14. Peiffer DA, Le JM, Steemers FJ, et al. *Genome Res*, **16(9)**:1136-48 (2006).
15. Gunderson KL, Steemers FJ, Lee G, et al. *Nat Genet*, **37(5)**:549-54 (2005).
16. Lindblad-Toh K, Tanenbaum DM, Daly MJ, et al. *Nat Biotechnol*, **18(9)**:1001-5 (2000).
17. Bolstad BM, Irizarry RA, Astrand M, et al. *Bioinformatics*, **19(2)**:185-93 (2003).
18. Lin M, Wei LJ, Sellers WR, et al. *Bioinformatics*, **20**:1233-40 (2004).
19. Laframboise T, Harrington D, Weir BA. *Biostatistics*, **8(2)**:323-36 (2007).
20. Bengtsson H, Irizarry R, Carvalho B, et al. *Bioinformatics*, **24(6)**:759-67 (2008).
21. Korn JM, Kuruvilla FG, McCarroll SA, et al. *Nat Genet*, **40(10)**:1253-60 (2008).
22. Zhao X, Weir BA, LaFramboise T, et al. *Cancer Res*, **65(13)**:5561-70 (2005).
23. Olshen AB, Venkatraman ES, Lucito R, et al. *Biostatistics*, **5(4)**:557-72 (2004).
24. Venkatraman ES, Olshen AB. *Bioinformatics*, **23(6)**:657-63 (2007).
25. Polzehl J, Spokoiny S. *J R Stat Soc, Ser B*, **62(2)**:335–354 (2000).
26. Hupé P, Stransky N, Thiery JP, et al. *Bioinformatics*, **20(18)**:3413-22 (2004).
27. Affymetrix File Parsing SDK [http://www.bioconductor.org/packages/2.2/bioc/html/affxparser.html].
28. McCarroll SA, Kuruvilla FG, Korn JM, et al. *Nat Genet*, **40(10)**:1166-74 (2008).
29. Kirkpatrick S, Gelatt CD Jr, Vecchi MP. *Science*, **220(4598)**:671-680 (1983).
30. dChip Software Website [http://www.dchip.org].
31. Li C, Wong WH. *PNAS*, **98**:31-6 (2001).
32. Pinto D, Marshall C, Feuk L, et al. *Hum Mol Genet*, **16 Spec No. 2**:R168-73 (2007).
33. Database of Genomic Variants [http://projects.tcag.ca/variation/].
34. Schouten JP, McElgunn CJ, Waaijer R, et al. *Nucleic Acids Res*, **30(12)**: e57 (2002).
35. LaFramboise Lab Software Website [http://mendel.gene.cwru.edu/laframboiselab/software.php].