Optimization Algorithms for Identification and Genotyping of Copy Number Polymorphisms in Human Populations

Gökhan Yavaş[1], Mehmet Koyutürk[1,3], and Thomas LaFramboise[2,3]

[1] Department of Electrical Engineering & Computer Science,
Case Western Reserve University, Cleveland, OH, USA
[2] Department of Genetics, Case Western Reserve University, Cleveland, OH, USA
[3] Center for Proteomics & Bioinformatics, Case Western Reserve University,
Cleveland, OH, USA

Abstract. Recent studies show that copy number polymorphisms (CNPs), defined as genome segments that are polymorphic with regard to genomic copy number and segregate at greater than 1% frequency in the populations, are associated with various diseases. Since rare copy number variations (CNVs) and CNPs bear different characteristics, the problem of discovering CNPs presents opportunities beyond what is available to algorithms that are designed to identify rare CNVs. We present a method for identifying and genotyping common CNPs. The proposed method, POLYGON, produces copy number genotypes of the samples at each CNP and fine-tunes its boundaries by framing CNP identification and genotyping as an optimization problem with an explicitly formulated objective function. We apply POLYGON to data from hundreds of samples and demonstrate that it significantly improves the performance of existing single-sample CNV identification methods. We also demonstrate its superior performance as compared to two other CNP identification/genotyping methods.

Keywords: CNV, CNP, optimization.

1 Introduction

Genetic differences that can be identified with single nucleotide polymorphism (SNP) microarrays include SNPs [1] and copy number variants (CNVs) [2]. CNVs are defined as chromosomal segments of at least 1000 bases (1 kb) in length that vary in number of copies from human to human. To date, several methods have been proposed for inferring CNVs from SNP array data [3-6]. In a recent study [7], we have formulated CNV identification as an optimization problem with an explicitly designed objective function that is characterized by several adjustable parameters. Our method, ÇOKGEN, efficiently identifies CNVs using a variant of the well-known simulated annealing heuristic.

All of these approaches are specifically designed for identifying rare or *de novo* CNVs by individually searching a sample's genome for regions in which evidence of copy number deviation exists. On the other hand, recent genome-wide association

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 74–85, 2010.

studies (GWAS) have underscored the importance of identifying common CNPs, associating them with several complex disease phenotypes [8-11]. Although these results highlight the need for dedicated methods for common CNP identification, most of the methods for CNV identification have not yet separated the ideas of identification and genotyping of common CNPs from discovery of rare CNVs.

In this paper, we present a method for identifying common copy number polymorphisms. The proposed method, POLYGON, takes as input the copy number variants identified by a single-sample CNV identification algorithm (e.g., ÇOKGEN [7], PennCNV [6], Birdseye [4]) and implements a computational framework to (i) identify CNVs in different samples that might correspond to the same variant in the population (candidate CNPs), (ii) adjust the boundaries of these candidate CNPs by drawing strength from raw copy number data from multiple samples, and (iii) determine copy number genotypes in the study. The key ingredient of this computational framework is an explicitly formulated objective function that takes into account several criteria, which are carefully designed to quantify the desirability of a CNP genotype with respect to various biological insights and experimental considerations. Namely, these criteria include minimizing variability in raw copy numbers of markers that are assigned to the same copy number class across samples, and maximizing raw copy number differences between samples that are assigned different copy numbers. We then develop algorithms that find copy number genotypes that optimize this function for fixed boundaries, and use this algorithm in a hierarchical manner to precisely adjust the boundaries of each CNP. Our performance analysis shows that POLYGON dramatically improves the performance of single sample methods in terms of Mendelian concordance and provides a moderate improvement in terms of sensitivity. Furthermore, we demonstrate its superior performance when compared to two other recurrent CNP detection algorithms presented in [12].

In the next section, we describe the general algorithmic framework for POLYGON, formulate CNP identification and genotyping as an optimization problem and present algorithms to solve this problem. Subsequently, in Section 3, we provide comprehensive experimental results on the performance of POLYGON in inferring CNPs from CNVs identified by three state-of-the-art CNV identification algorithms; ÇOKGEN, PennCNV, and Birdseye. We also compare the performance of our method to two other multi sample methods, COMPOSITE and COVER [12]. Finally, in Section 4, we discuss these results.

2 Methods

POLYGON first uses an existing algorithm to identify CNVs in each sample. The output of this step generates a list of CNVs for each sample, which may correspond to CNPs, rare/*de novo* CNVs, or false positives. Copy number genotypes for these CNVs are not required by POLYGON. Subsequently, POLYGON reconciles these CNVs in two phases:

(i) Clustering of identified CNVs to obtain an initial set of *candidate CNPs* (clusters of CNVs that potentially correspond to the same event).
(ii) Fine tuning of the boundaries of candidate CNPs and precise estimation of number of copies in each sample.

In the remainder of this section, we explain the algorithmic details of these two phases.

2.1 Problem Definition

Consider a study in which a set N of samples are screened via SNP microarray technology to obtain raw copy number estimates for a set M of markers on a single chromosome (we formulate the problem in the context of a single chromosome since each chromosome can be processed separately). The HapMap [1] dataset contains 270 samples and a total of approximately 1.8 million markers (for Affymetrix 6.0 SNP array) over 23 chromosomes. The objective of the CNP identification and genotyping problem is to assign a copy number to all markers in all samples such that copy number assignment is smooth across markers and consistent across samples. Formally, we are seeking a mapping S: N x M → C, where C = {0, 1¸ 2, 3, 4} denotes the set of possible copy numbers and 0, 1¸ 2, 3, and 4, respectively denote homozygous deletion, hemizygous deletion, normal copy number, hemizygous duplication, and homozygous duplication (some samples may contain more than four copies, but all such cases are encapsulated into copy number class 4 to have a compact set of copy number classes). To find the mapping, POLYGON uses two data types:

(i) The set V = $\{v_1, v_2, \ldots v_K\}$ of CNV calls provided by a single-sample algorithm. Each CNV $v \in$ V is a pair (s_v, e_v) where s_v and e_v denote the start and end markers of the region v, and $M_v=\{i\text{: } s_v \leq i \leq e_v\}$ defines the set of markers flanked by the pair. The length of CNV v is defined as $l_v = |M_v| = e_v - s_v + 1$.
(ii) For each sample marker $(n, m) \in$ N x M, the raw copy number estimate $R_{n,m}$. These estimates are also provided by the single-sample algorithms which are utilized for CNV identification.

POLYGON implements a two-phase algorithm to call CNPs from these raw copy numbers and initial set of CNVs. The aim of the first phase is to obtain a set, W =$\{w_1, w_2, .., w_t\}$, of candidate CNPs by clustering CNVs identified on different samples according to their chromosomal coordinates. Each candidate $w \in$ W is defined by the pair (s_w, e_w) where s_w and e_w represent the start and end markers of the region. Similar to M_v, $M_w=\{i\text{: } s_v \leq i \leq e_v\}$ defines the set of markers in CNP w. Based on the definition of w, we reduce the CNP genotyping problem to finding a set of functions S_w: N → C for all $w \in$ W where S_w determines the genotype of each sample at CNP w. Then, for each $(n, m) \in$ N x M**,** $S(n, m)$ is defined as $S_w(n)$ if $m \in M_w$ and 2 otherwise for all $w \in$ W.

Thus, in the second phase, we utilize an optimization based framework to find the optimal S_w for each $w \in$ W (hence we obtain the optimal genotyping of all CNPs which implies optimal S), while fine-tuning its boundaries.

2.2 Identification of Candidate CNPs

In the first phase, POLYGON clusters individual CNVs based on the start and end markers to obtain the candidate CNPs that represent "similar" CNVs on different samples. To assess the similarity between two CNVs, we use the *minimum reciprocal overlap* (*MRO*) measure. For two CNVs v_1 and v_2, let $o(v_1,v_2) = |M_{v_1} \cap M_{v_2}|$ denote the

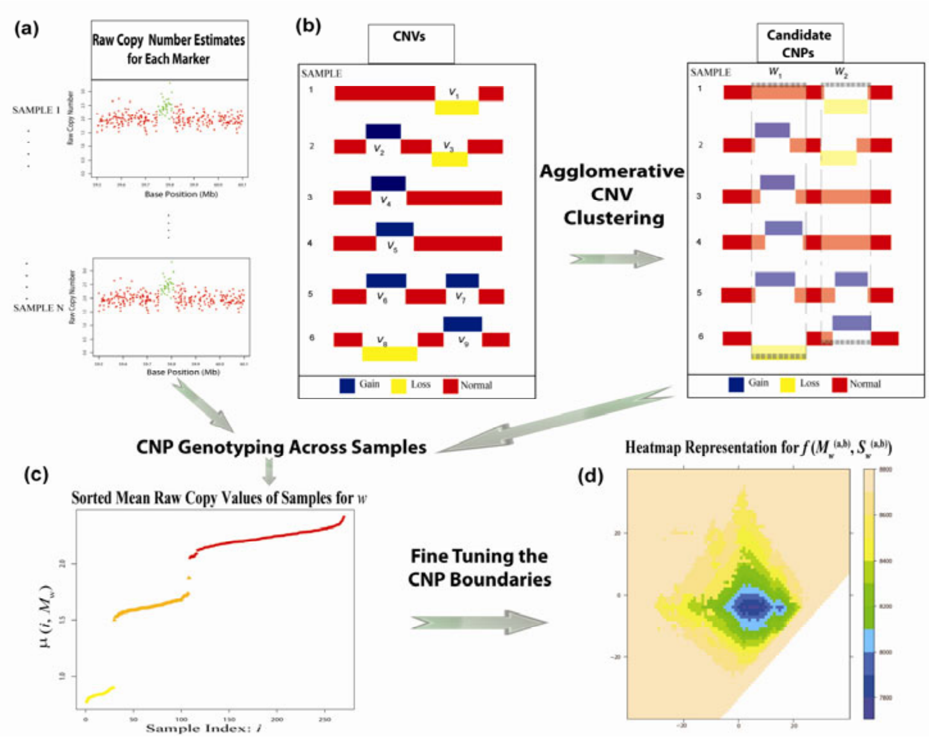
(a)
Raw Copy Number Estimates for Each Marker
SAMPLE 1
SAMPLE N
Base Position (Mb)
(b)
CNVs
SAMPLE
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
Gain
Loss
Normal
Agglomerative CNV Clustering
Candidate CNPs
w_1
w_2
CNP Genotyping Across Samples
(c)
Sorted Mean Raw Copy Values of Samples for w
$\mu(i, M_w)$
Sample Index: i
Fine Tuning the CNP Boundaries
(d)
Heatmap Representation for $f(M_w^{(a,b)}, S_w^{(a,b)})$

Fig. 1. Algorithmic workflow of POLYGON. (a) The raw copy estimates as provided by the single-sample CNV detection algorithms. (b) Our agglomerative CNV clustering algorithm takes as input the CNVs identified by the single-sample CNV detection algorithms, to obtain a set of candidate CNPs. Here, the algorithm is illustrated on a toy example set of CNVs, V = $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, obtaining the set of candidate CNPs W=$\{w_1, w_2\}$. (c) For each $w \in$ W, to obtain the optimal copy number genotyping in each sample for given candidate boundaries of *w,* the samples are sorted with respect to average copy number within these boundaries. Subsequently, high gradient points in this ordering are identified to segregate samples into copy number classes. The sorted mean raw copy numbers and the associated genotypes are for a real *w* identified by POLYGON in the HapMap dataset, and are not related to the toy example of (b). The samples genotyped with copy number classes 0, 1 and 2 are shown with colors yellow, orange and red, respectively. (d) The heat map displays the matrix colored according to the values of the objective function $f(M_w^{(a,b)}, S_w^{(a,b)})$ at the optimal genotype solution for each candidate boundary (a,b) as computed by the procedure in (c). Note that the coordinates on the horizontal and vertical axis correspond to the start and end coordinates of candidate boundaries for *w*, and that for demonstration purposes they have been re-centered so that the initial boundaries are at (0,0). Once this heatmap is obtained, the optimal boundaries of the CNP are set to (a, b) that correspond to the minimum value in this matrix and the copy number genotypes are given by the optimal assignment for those boundaries (as computed in (c)).

size of the overlap between v_1 and v_2. Then the minimum reciprocal overlap of v_1 and v_2 is defined as

$$MRO\ (v_1, v_2) = \min\left(\frac{o(v_1, v_2)}{l_{v_1}}, \frac{o(v_1, v_2)}{l_{v_2}}\right).$$

Using this similarity measure, POLYGON agglomeratively clusters CNVs using a conservative complete-linkage based criterion to measure the similarity between groups of CNVs. We use $\Pi = \{\rho_1, \rho_2, ..., \rho_t\}$ to denote a set of CNV clusters where each $\rho_i \in \Pi$ represents a set of CNVs. At the beginning of clustering, each CNV constitutes a cluster by itself, i.e., $\Pi^{(0)} = \{\{v_i\}: v_i \in \mathbf{V}\}$. At each iteration, two candidate CNV clusters with maximum similarity are merged, where the similarity between CNV clusters ρ_i and ρ_j is defined as

$$MRO\ (\rho_i, \rho_j) = \min_{\mathrm{v}_\mathrm{q} \in \rho_\mathrm{i}, \mathrm{v}_\mathrm{p} \in \rho_\mathrm{j}} \{MRO\ (v_q, v_p)\}.$$

This process continues until the similarity between any two clusters goes below a predefined threshold. The set obtained through the clustering process $\Pi = \{\rho_1, \rho_2, ..., \rho_t\}$ is then used to obtain the candidate CNP set $\mathbf{W} = \{w_1, w_2, .., w_t\}$, where each $w_i = (s_{w_i}, e_{w_i})$ and $s_{w_i} = \min_{v \in \rho_i}\{s_v\}$ and $e_{w_i} = \max_{v \in \rho_i}\{e_v\}$. In this study, we have chosen the overlap threshold as 0.5, which guarantees that all the CNVs that correspond to a single candidate CNP have at least 50% mutual overlap in terms of markers that they span. Note that we do not take into consideration the type of the CNV (*e.g.,* deletion *vs.* insertion) while clustering CNVs. Therefore, it is possible that a loss and a gain can be represented by the same candidate CNP as long as they share at least 50% of their markers. The motivation behind this approach is that both gain and loss events were reported for the same region in different samples in previous research [13]. In Figure 1(b), this process is illustrated with a toy example.

The next phase of POLYGON processes each candidate CNP individually and determines the CNP genotype of each sample, while fine tuning its boundaries.

2.3 Identifying CNP Genotypes and Fine-Tuning of CNP Boundaries

Once the set of candidate CNPs are obtained, for each CNP region *w*, we select a window of markers to be searched exhaustively to fine-tune the boundaries of *w*. The initial boundaries of the window containing *w* are extended to allow consideration of the markers bordering initially identified *w* for enlarging, shrinking or shifting its markers. We define the search window for $w \in \mathbf{W}$ as the set of markers $\Omega_w = \{i: s_w - \lceil l_w/2 \rceil \leq i \leq e_w + \lceil l_w/2 \rceil\}$.

In order to assess the quality of the boundaries of a CNP and the genotype calls in each sample, we formulate an objective function that brings together multiple quantitative criteria that gauge the suitability of CNP genotype calls based on observed array intensities of all the samples. This objective function takes into account the smoothness of raw copy number estimates over contiguous markers that are declared to have identical copy numbers, as well as consistency of genotype calls of the same CNP across samples.

We define objective function $f\,(M_w, S_w)$ as a combination of the following objective criteria:

- **Variation in raw copy numbers within each copy number class should be minimized.** Ideally, the raw copy number estimates (i.e., $R_{n,m}$) for markers that are assigned identical copy numbers should be similar. For a given CNP w and copy number assignment S_w, let the set of samples assigned to class $c \in$ C be $\Psi(c) = \{n \in \mathrm{N} : S_w(n) = c\}$. The mean raw copy number for class c can be computed as follows:

$$\mu(c) = \begin{cases} \dfrac{\sum\limits_{n \in \Psi(c)} \sum\limits_{m \in \Omega_w} R_{n,m} + \sum\limits_{n \in N \setminus \Psi(c)} \sum\limits_{m \in \Omega_w \setminus M_w} R_{n,m}}{|\Omega_w||\Psi(c)| + |\Omega_w \setminus M_w||N \setminus \Psi(c)|} & \text{if c = 2} \\[2ex] \dfrac{\sum\limits_{n \in \Psi(c)} \sum\limits_{m \in M_w} R_{n,m}}{| M_w ||\Psi(c)|} & \text{otherwise} \end{cases} .$$

The mean raw copy number values for aberrant copy number classes are simply calculated by averaging the raw copy estimates in region M_w across all samples genotyped with the specified copy number class. However, for the "normal" copy number class, this computation is slightly more complicated since the markers in all samples that are outside the boundaries of w also contribute to the mean of the "normal" copy number class. Then, the total intra-class variability induced by S_w is given by

$$\sigma(M_w, S_w) = \sum_{c \in C \setminus 2} \sum_{n \in \Psi(c)} \left(\sum_{m \in M_w} \left| R_{n,m} - \mu(c) \right| + \sum_{m \in \Omega_w \setminus M_w} \left| R_{n,m} - \mu(2) \right| \right) + \sum_{n \in \Psi(2)} \sum_{m \in \Omega_w} \left| R_{n,m} - \mu(2) \right|.$$

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

- **Variation in raw copy numbers across different copy number classes should be maximized.** The criterion formulated above focuses on the internal variation in a copy number class. However, it is also important to accurately separate different copy number classes from each other, since the number of variants in the sample is unknown and intra-class variation can be minimized by artificially increasing the number of genotype classes across samples. For this reason, we formulate an objective criterion that penalizes excessive copy number classes. Formally, we define

$$\chi(M_w, S_w) = \sum_{c=0}^{3} 2^{\frac{1}{\mu(c+1) - \mu(c)}} I(|\Psi(c)||\Psi(c+1)| \neq 0)$$

as an objective criterion to be minimized. Here $I(.)$ denotes the indicator function (*i.e.*, it is equal to 1 if the statement being evaluated is true, and 0 otherwise). Observe that this function grows exponentially with the reciprocal of the difference between the mean raw copy numbers of markers assigned to consecutive copy number classes, and is therefore minimized when similar raw copy numbers are assigned to the same class.

• **Filtering out noise by eliminating smaller regions.** Longer CNPs 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 CNP length as an additional objective criterion. We then define $\lambda(M_w) = \frac{1}{2^{l_w}}$ as an objective criterion that penalizes shorter CNPs.

• **The optimal CNP identification and genotyping problem.** We use a linear combination of the criteria above as an objective function to assess the quality of a CNP region and assignment of copy number genotypes. Namely, for a given candidate CNP *w,* an assignment of markers M_w to *w*, and assignment S_w of copy numbers to these markers in each sample is defined as

$$f(M_w, S_w) = k_\sigma \sigma(M_w, S_w) + k_\chi \chi(M_w, S_w) + k_\lambda \lambda(M_w)$$

The objective of the CNP identification and genotyping problem is to find M_w and S_w that together minimize $f(M_w, S_{w.})$. Here, the tunable coefficients $k_\sigma, k_\chi, k_\lambda$ adjust the relative importance of the objective criteria with respect to each other. In our experiments, we use a prohibitively large value for k_λ to eliminate CNP instance calls on smaller regions that are likely to be false positives. The parameters k_σ and k_χ are used to adjust the apparent trade-off between the intra-class and the inter-class variation. Without loss of generality, we require that $k_\sigma + k_\chi = 1$ so that the parameters can be adjusted in an interpretable way. For our experimental evaluations reported in this paper, we use $k_\sigma = 0.5$ and $k_\chi = 0.5$. Note also that, for a given M_w and S_w, the computation of $f(M_w, S_w)$ requires $O(|\mathbf{N}||\Omega_w|)$ time.

2.4 Algorithms for Optimal CNP Identification and Copy Number Genotyping

We now describe the algorithm we use to find the objective function minimum, thereby solving the CNP identification and genotyping problem. A solution to a given instance of the problem is characterized by assignment of marker boundaries to the CNP (M_w) along with the copy number genotyping $S_w(n)$ for each sample $n \in$ N. Consequently, an optimal solution to the problem can be determined by finding an optimal S_w for each possible M_w and choosing the best among these solutions across all possible assignments of M_w. Since a CNP region is by definition composed of contiguous markers and the problem is defined within a fixed segment of markers Ω_w, there are $|\Omega_w|(|\Omega_w|+1)/2$ possibilities for M_w, making such an exhaustive search feasible. Motivated by this insight, we now discuss how an optimal assignment of S_w can be found for fixed M_w.

(i) Optimal CNP genotyping for fixed CNP boundaries. When the boundaries of the CNP are fixed, the solution to the CNP genotyping problem is uniquely determined by the assignment of each sample to a copy number class for the CNP region at hand. To find an optimal solution to this problem, POLYGON uses a top-down approach that starts from a conservative solution that assigns all samples to the same class and iteratively improves this solution by dividing samples into separate classes as necessary. Initially, all samples are assigned to the "normal" class, *i.e.*,

$S_w^{(0)}(n) = 2$ for all $n \in$ N. At each step of the algorithm, samples that are assigned to the same copy number class are iteratively considered to check whether it is possible to further improve the solution by dividing this partition of samples into two sub partitions with different copy number classes. To find the best possible partitioning of the samples in a group, we use the mean raw copy number of markers within M_w on each sample, computed as:

$$\mu(n, M_w) = \frac{\sum_{m \in M_w} R_{n,m}}{l_w} \quad .$$

Assume, without loss of generality, that the samples are ordered according to $\mu(n, M_w)$. That is, $\mu(n, M_w) \leq \mu(n+1, M_w)$ for all $i = 1, \ldots, |\mathrm{N}|-1$. The aim of our algorithm is to divide the ordered of set samples in up to five partitions such that each partition corresponds to the set of samples with a copy number class and the objective function f is minimized for the given class assignments. It can be shown that the optimal copy number genotype assignment must preserve the $\mu(n, M_\mathrm{w})$ ordering. Based on this observation, we develop a heuristic based on the notion that a sample at which the copy number genotype change is most likely to happen is the one at which the maximum increase is observed in between $\mu(n, M_\mathrm{w})$ and $\mu(n+1, M_\mathrm{w})$ values.

Our algorithm is executed using a series of splits dividing one copy number class into two at each stage. Let $S_w^{(i)}$ denote the solution after the i^{th} split where $0 \leq i \leq 4$ (since there can be at most 5 copy number class partitions) and $\Psi^{(i)}(c)$ denotes the set of samples in the partition for copy number class $c \in$ C after the i^{th} split. In each round, our algorithm introduces a new copy number class partition by *splitting* an already existing copy number class partition c. This is done by choosing a sample n^*, and then either moving all samples $n \leq n^*$ in n^*'s copy number class c to copy number class c-1, or moving all samples $n > n^*$ in n^*'s copy number class to copy number class c+1. We call n^* a split sample. However, if the algorithm tries to split a copy number class partition by re-introducing an already existing copy number class partition (*i.e.,* if copy number c-1 or c+1 is already assigned to some samples), this split becomes invalid and our algorithm tries another n^* for this round of split procedure. Let $\mathrm{Q}^{(i)}$ denote the set of candidate split samples, *i.e*., samples that are not used in one of the previous splits or are skipped by the algorithm . Initially, we have $S_\mathrm{w}^{(0)}(n) = 2$ for all $n \in$ N , $\Psi^{(0)}(2) = \mathrm{N}$ and $\Psi^{(0)}(c) = \varnothing$ for $c \in \mathrm{C} \setminus 2$, and $\mathrm{Q}^{(0)} = \mathrm{N}$.

For each sample $1 \leq n \leq |\mathbf{N}|-1$, let $\Delta(n) = \mu(n+1, M_\mathrm{w}) - \mu(n, M_\mathrm{w})$ denote the gradient of mean copy numbers at sample n. At each round of the algorithm, the sample $n^* = \mathrm{argmax}_{n \in \mathrm{Q}^{(i)}}\{\Delta(n)\}$ is selected as the splitting sample, since it would yield the highest inter-class variance for the new class partitions being created. Assume that n^* is assigned copy number c at this point. One of the sub-partitions that can be obtained by splitting the partition c will obviously be the old partition c. In order to determine whether the other sub-partition will be c-1 or c+1, we check the similarity of the mean raw copy number of each sub-partition to that of the original partition. To do so, the mean raw copy number for each sub-partition is computed as:

$$\mu_c = \frac{\sum_{j \in \Psi^{(i)}(c)} \mu(j, M_w)}{|\Psi^{(i)}(c)|} \;, \quad \mu_c' = \frac{\sum_{j \in \Psi^{(i)}(c), j \leq n^*} \mu(j, M_w)}{n^* - \min(\Psi^{(i)}(c)) + 1} \;, \quad \mu_c'' = \frac{\sum_{j \in \Psi^{(i)}(c), j > n^*} \mu(j, M_w)}{\max(\Psi^{(i)}(c)) - n^*} \;.$$

There are two cases to be considered.

Case 1: $\left|\mu_c - \mu_c'\right| \leq \left|\mu_c - \mu_c''\right|$

In this case, the samples in the lower sub-partition have more similar mean raw copy number to that of the samples in the original partition. Therefore, the newly introduced copy number class partition should be *c*+1 and the samples from min(Ψ(*c*)) to *n** will remain in partition *c* and samples from *n**+1 to max(Ψ(*c*)) will be assigned to partition *c*+1 in the new solution, i.e.,

$$S_w^{(i+1)}(n) = \begin{cases} c+1 & \text{for } n \in \Psi^{(i)}(c) \text{ and } \quad n > n^* \\ S_w^{(i)}(n) & \text{otherwise} \end{cases}.$$

Case 2: $\left|\mu_c - \mu_c'\right| > \left|\mu_c - \mu_c''\right|$

In this case, the upper sub-partition is more similar to the original partition in terms of mean raw copy number. Thus, the newly introduced copy number class partition should be *c*-1 and the samples from *n**+1 to max(Ψ(*c*)) will be assigned to class *c* and samples from min(Ψ(*c*)) to *n** will be assigned to class *c*-1 in the new solution, i.e.,

$$S_w^{(i+1)}(n) = \begin{cases} c-1 & \text{for } n \in \Psi^{(i)}(c) \text{ and } \quad n \leq n^* \\ S_w^{(i)}(n) & \text{otherwise} \end{cases}.$$

Note that splits in cases 1 and 2 are invalid if $\Psi^{(i)}(c+1) \neq \emptyset$ and $\Psi^{(i)}(c-1) \neq \emptyset$, respectively (i.e., the split is trying to introduce a copy number class partition that already exists). In that case, the algorithm updates the set of candidate split samples as $Q^{(i)} = Q^{(i)} \setminus n^*$, and repeats the procedure for finding a split sample for the current $S_w^{(i)}$ as described above. In the case of a valid split, it checks whether the new solution $S_w^{(i+1)}$ improves the current solution $S_w^{(i)}$ in terms of the objective function (i.e., if $f(M_w, S_w^{(i+1)}) < f(M_w, S_w^{(i)})$). If so, the algorithm sets $Q^{(i+1)} = Q^{(i)} \setminus n^*$, updates $\Psi^{(i+1)}$ according to $S_w^{(i+1),}$ and moves to the next splitting round. The algorithm will stop if the number of copy number class partitions reaches five, the set of candidate split samples becomes empty (i.e., $Q^{(i)} = \emptyset$), or the new solution $S_w^{(i+1)}$ does not improve the current solution $S_w^{(i)}$ in terms of the objective function. In these cases, $S_w^{(i)}$ is reported as the optimal solution. Note that the running time of this algorithm is $O(|N||\Omega_w|)$, since the dominant computation throughout the course of the algorithm is the computation of *f* for a constant number of times.

In Figure 1(c), for a CNP *w*, the ordered samples and the corresponding mean raw copy numbers $\mu(n, M_w)$ for each sample $n \in \{1, 2,.., 270\}$ are shown. As evident in the plot, the top candidate split samples are those where the biggest jumps occur between consecutive μ values. After applying the above procedure, we find that the CNP *w* manifests itself in three different copy number classes across the sample set **N**. The samples genotyped with copy number 0, 1 and 2 classes are colored with yellow, orange and red, respectively.

(ii) Finding the optimal boundaries of a candidate CNP. The above procedure gives a solution to the optimal CNP assignment problem for fixed CNP boundaries (M_w). Recall that for each CNP *w*, an initial estimate of its boundaries is available from the first phase of POLYGON. We exhaustively search all possible sub-windows

$[a, b]$ within Ω_w (where $s_w - \lceil l_w/2 \rceil \leq a \leq b \leq e_w + \lceil l_w/2 \rceil$), finding the optimal CNP genotyping $S_w^{(a,b)}$ for each candidate boundary $M_w^{(a,b)}$. Finally, the $M_w^{(a,b)}$ and $S_w^{(a,b)}$ that minimize $f(M_w^{(a,b)}, S_w^{(a,b)})$ are returned as the optimal CNP assignment for CNP w. This procedure is illustrated in Figure 1(d). From the heat map in the figure, it can be observed that the optimal boundaries obtained after this method is applied are different from the initial boundaries of w. The total runtime of this algorithm is $O(|N||\Omega_w|^3)$, which is reasonable in practical cases since the size of region Ω_w does not exceed several hundred markers for majority of the CNPs discovered by our method.

3 Results

We apply our algorithm to Affymetrix 6.0 SNP array data from 270 HapMap individuals. We use three different algorithms, ÇOKGEN [7], PennCNV [6] and Birdseye [4] to detect the initial set of CNVs that serve as input to POLYGON.

3.1 Methods Used for Comparison

There are few CNP identification methods available for SNP array platforms. Here we compare POLYGON with two methods, COMPOSITE and COVER, which were published quite recently [12]. Similar to POLYGON, these two methods use CNVs identified by other methods to call common CNPs. Thus, they utilize the same type of data (CNVs mined on the Affymetrix 6.0 SNP array by PennCNV and an annotation file containing the genomic coordinates of the markers) and produce the same type of output with POLYGON. It should be noted that there exists another method, Canary [4], for genotyping CNPs. However, it is designed to genotype the CNP maps given by [13] and is not a CNP discovery method *per se*. For this reason, we do not include Canary in our comparisons.

To simplify the discordance and sensitivity analysis and to be consistent with the results of the single-sample based CNV identification algorithms, a CNP genotyped by POLYGON, COMPOSITE or COVER is treated as a single gain or loss CNV event in the analyses reported here. For the discordance and sensitivity analysis, we use the *MRO* measure (as defined in Section 2.2) with a threshold of 0.5 to decide whether two CNVs identified in two different individuals correspond to the same event.

3.2 Trio Discordance Comparison across Methods

The 60 mother-father-child trios in the HapMap data set were used to assess the accuracy of CNV genotyping algorithms by measuring the rate of Mendelian concordance. A gain or loss 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.

For all of the single-sample CNV identification methods, POLYGON greatly improves trio discordance. POLYGON reduces ÇOKGEN's trio discordance from 30.8% to 20.1%. Similarly, it reduces PennCNV's trio discordance from 32.9% to 16.2%. On the other hand, both COMPOSITE and COVER reduce PennCNV's trio discordance rate to around 26%. These results demonstrate the superior ability of POLYGON for CNP identification and copy number genotyping across samples.

3.3 Sensitivity Comparison across Methods

A recent study [14] 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 to be harbored in at least one of the 270 HapMap individuals. We use this as a "gold standard" data set on which to evaluate the sensitivity of our method.

POLYGON improves the sensitivity of two single-sample based CNV identification methods. While ÇOKGEN achieves a sensitivity of 86%, POLYGON improves this to 88.3%. Similarly, sensitivity increases from 84.7% to 89.9% when POLYGON is run with CNVs obtained by Birdseye. Interestingly, on the other hand, PennCNV and POLYGON on PennCNV achieve the same sensitivity rate of 88.6%. These figures are clearly superior to the sensitivity of both COMPOSITE (62.8%) and COVER (40.2%).

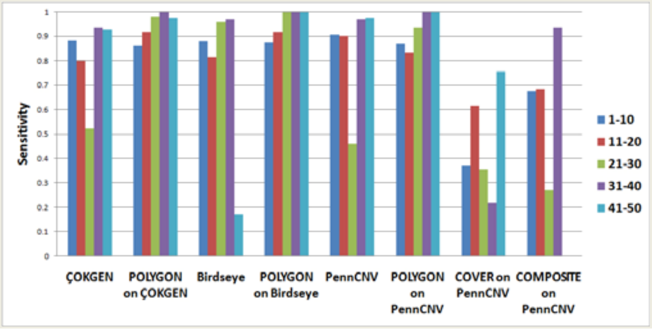

Fig. 2. Sensitivity of different algorithms. Each bar represents the sensitivity of the associated method in the specified frequency stratum.

In Figure 2, we compare the sensitivity of the methods stratified by the gain/loss frequencies of the CNVs. The purpose of this analysis is to see whether an algorithm that explicitly targets common CNPs is more successful in calling common CNPs accurately (as compared to rare CNVs). Indeed, as seen in the figure, POLYGON improves the sensitivity of all CNV identification methods for gains/losses existing in more than 20 samples, demonstrating that POLYGON is well-suited to detect common CNPs. Furthermore, for gains/losses that occur in at least 30 samples, POLYGON consistently achieves sensitivity above 98%, regardless of the algorithm that is used to identify the initial set of CNVs. This observation suggests that POLYGON is also quite robust against changes in the input set of CNVs.

4 Conclusion

We have presented a method to detect and genotype germline copy number polymorphisms (CNPs) from SNP array data and a set of CNVs. Our approach will be useful for researchers querying constitutional DNA for association of CNP alleles with disease. Indeed, CNPs are emerging as important factors in a growing number of diseases. POLYGON's ability to identify recurrent variants is particularly crucial in GWAS, as variations frequently observed in a significant proportion of the population may have a significant impact on human disease.

The current work shows that the problem of detecting CNPs 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. With detailed experimental studies on the HapMap dataset, we have demonstrated its sensitivity to identify especially common CNPs, while keeping a low false positive rate, as demonstrated by high Mendelian consistency in trios.

References

1. IHMC: A haplotype map of the human genome. Nature 437, 1241–1242 (2005)
2. Feuk, L., et al.: Structural variation in the human genome. Nat. Rev. Genet. 7, 85–97 (2006)
3. Colella, S., et al.: QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. Nucleic Acids Res. 35, 2013–2025 (2007)
4. Korn, J.M., et al.: Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs. Nat. Genet. 40, 1253–1260 (2008)
5. Olshen, A.B., et al.: Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5, 557–572 (2004)
6. Wang, K., et al.: PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. Genome Res. 17, 1665–1674 (2007)
7. Yavaş, G., et al.: An optimization framework for unsupervised identification of rare copy number variation from SNP array data. Genome Biology 10, R119 (2009)
8. Gonzalez, E., et al.: The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. Science 307, 1434–1440 (2005)
9. Aitman, T.J., et al.: Copy number polymorphism in Fcgr3 predisposes to glomerulonephritis in rats and humans. Nature 439, 851–855 (2006)
10. Fanciulli, M., et al.: FCGR3B copy number variation is associated with susceptibility to systemic, but not organ-specific, autoimmunity. Nat. Genet. 39, 721–723 (2007)
11. Yang, Y., et al.: Gene copy-number variation and associated polymorphisms of complement component C4 in human systemic lupus erythematosus (SLE): low copy number is a risk factor for and high copy number is a protective factor against SLE susceptibility in European Americans. Am. J. Hum. Genet. 80, 1037–1054 (2007)
12. Shu Mei, T., et al.: Identification of recurrent regions of copy-number variants across multiple individuals. BMC Bioinformatics 11, 147 (2010)
13. McCarroll, S.A., et al.: Integrated detection and population-genetic analysis of SNPs and copy number variation. Nat. Genet. 40, 1166–1174 (2008)
14. Pinto, D., et al.: Copy-number variation in control population cohorts. Hum. Mol. Genet. 16, R168–R173 (2007)