Genome Biology

This Provisional PDF corresponds to the article as it appeared upon acceptance. Copyedited and fully formatted PDF and full text (HTML) versions will be made available soon.

An Optimization Framework for Unsupervised Identification of Rare Copy Number Variation from SNP Array Data

Genome Biology 2009, **10**:R119 doi:10.1186/gb-2009-10-10-r119

Gokhan Yavas (gokhan.yavas@case.edu)
Mehmet Koyuturk (koyuturk@eecs.case.edu)
Meral Ozsoyoglu (meral@case.edu)
Meetha Gould (meetha.gould@case.edu)
Thomas LaFramboise (thomas.laframboise@case.edu)

ISSN 1465-6906

Article type Method

Submission date 21 August 2009

Acceptance date 22 October 2009

Publication date 22 October 2009

Article URL http://genomebiology.com/2009/10/10/R119

This peer-reviewed article was published immediately upon acceptance. It can be downloaded, printed and distributed freely for any purposes (see copyright notice below).

Articles in *Genome Biology* are listed in PubMed and archived at PubMed Central.

For information about publishing your research in *Genome Biology* go to

http://genomebiology.com/info/instructions/

© 2009 Yavas *et al.* , licensee BioMed Central Ltd.
This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An optimization framework for unsupervised identification of rare copy number variation from SNP array data

Gökhan Yavaş[1], Mehmet Koyutürk[1,3], Meral Özsoyoğlu[1], Meetha Gould[2], Thomas LaFramboise[2,3,4§]

[1]Department of Electrical Engineering & Computer Science, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH, 44106, USA

[2]Department of Genetics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH, 44106, USA

[3]Center for Proteomics and Bioinformatics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH, 44106, USA

[4]Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic Foundation, 9500 Euclid Avenue, Cleveland, OH 44195, USA

[§]Corresponding author

Email addresses:

GY: gokhan.yavas@case.edu

MK: koyuturk@eecs.case.edu

MÖ: meral@case.edu

MG: meetha.gould@case.edu

TL: thomas.laframboise@case.edu

Abstract

Copy number variants (CNVs) have roles in human disease, and DNA microarrays are important tools for identifying them. In this paper, we frame CNV identification as an objective function optimization problem. We apply our method to data from hundreds of samples, and demonstrate its ability to detect CNVs at a high level of sensitivity without sacrificing specificity. Its performance compares favorably with currently available methods and it reveals previously unreported gains and losses.

Background

Identifying DNA variants that contribute to disease is a central aim in human genetics research. 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]. 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 [5], Crohn's disease [6], autism [7], and schizophrenia [8]. 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 [9, 10], base composition [11, 12], 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. 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.

The Affymetrix SNP array was originally designed so that each SNP is interrogated by 24-40 unique probes. Of these, half are perfectly complementary to the sequence harboring the SNP site (perfect match probes), while half mismatch the sequence at the probe's middle nucleotide (mismatch probes). The mismatch probes were intended to capture background effects such as cross-hybridization. The perfect match/mismatch design was used for the 10,000, 100,000, and 500,000-SNP versions of the array. Most recently Affymetrix has introduced the SNP Array 6.0, which interrogates nearly one million SNPs and differs fundamentally from previous versions. First, each SNP on the 6.0 array is interrogated only by six or eight perfect match probes – three or four replicates of the same probe sequence for each of the two alleles. Therefore, intensity data for each SNP consist of three or four repeated pairs of measurements. Second, the SNP probe sets are augmented with nearly one million CNV probes, which are meant to interrogate regions of the genome that do not harbor SNPs, but which may be polymorphic with regard to copy number. Each such CNV site is interrogated by only one probe.

For the Affymetrix platform, the community has largely settled upon quantile normalization [13] as a simple, yet effective, normalization method. The next step, raw copy number extraction, typically entails fitting some model to raw probe intensity data [14-17]. Methods devoted to 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) [17, 18], circular binary segmentation [19, 20], and adapted weight smoothing [21, 22]. 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 present a general framework to call CNVs from raw copy number using optimization, 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. Indeed, for N markers and C CNV classes, the solution space of the optimal copy number assignment problem is of size $O(C^N)$. Exhaustively searching for the optimal solution is quite infeasible unless N becomes very small. In our case, $N \approx 1.8$ million, so we adapt a simulated annealing based algorithm that efficiently searches the solution space at near-interactive rates.

We note here the distinction between CNVs and copy number polymorphisms (CNPs). CNPs are defined to be copy number variants that are present, with identical boundaries (and are therefore likely identical-by-descent), in at least 1% of the human population [23]. Computationally, such higher-frequency polymorphisms present opportunities for detection that are not otherwise possible. A recent study [17] proposes separate methods to detect CNVs and CNPs, with the latter involving detecting correlations in raw copy numbers across samples. The current work is designed to address the problem of identifying rare and *de novo* CNVs, as it does not make use of multiple samples to convert raw copy number into CNV inferences.

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 the proposed method compares favorably with four methods that are currently publicly available. Furthermore, we experimentally validate, using laboratory techniques on genomic DNA, several CNVs newly discovered by our method. These results demonstrate the proposed method's potential to uncover human genetic variation that may be missed by other computational approaches.

The general framework described in this paper is implemented and freely available in a flexible, user-friendly R package ÇOKGEN*. ÇOKGEN works from the raw binary .CEL files produced by the Affymetrix protocol. It performs all of the steps in Figure 1, including 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.

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 16128 autosomal CNVs over all the samples, for an average of 60 CNVs per individual. Of the 16128 CNVs, 15369 are identified in multiple individuals. Figure 2 graphically displays all CNVs identified by our method. As expected, many common CNVs are located near the centromeres and telomeres which are known to harbor variably repetitive elements.

The distribution of the CNVs among different ethnicities in the population is presented in Table 1. It is well known that Asian and Caucasian populations are genetically less diverse than African population due to population bottlenecks. This is reflected in Figure 3, which shows a shifted frequency distribution in the YRI CNVs relative to the CEU and JPT/CHB CNVs.

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 [23]. 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. Therefore, while comparing the performance of algorithms according to trio discordance rate, we also account for the number of frequently called CNVs, as discussed in the next subsection.

In the current study, to decide whether two CNVs (of the same type – loss or gain), c_1 and c_2, from two different samples correspond to the same event, we use the concept of minimum reciprocal overlap. We first define $o(c_1,c_2)$ as the number of markers existing in both c_1 and c_2 and $l(c)$ as the number of markers in a CNV c. Minimum reciprocal overlap ($MRO(c_1, c_2)$) of c_1 and c_2 is defined as:

$$MRO(c_1, c_2) = \min\left(\frac{o(c_1, c_2)}{l(c_1)}, \frac{o(c_1, c_2)}{l(c_2)} \right)$$

This measure provides a standard way of determining the similarity in the chromosomal location of two CNVs, regardless of the scale of the events. For our discordance and sensitivity analysis, we use the *MRO* measure with a threshold of 0.5 to decide whether two CNVs identified in two different individuals correspond to the same event. That is, at least half of c_1 must be overlapping with c_2 and vice versa for c_1 and c_2 to be considered as the same CNV in different samples.

Performance of ÇOKGEN in Comparison to Existing Software

We compared the performance of our algorithm with that of four other software packages. The *DNA-Chip Analyzer (dChip)* [24] is a Windows software package for Affymetrix platform and high-level analysis of gene expression microarrays and SNP microarrays [14, 25]. *Birdseye* [17] is a rare CNV identification tool based on HMMs, and is part of the Birdsuite platform [17]. *QuantiSNP* [26] is an analytical tool for the analysis of copy number variation using whole genome SNP genotyping data. It is originally developed for Illumina arrays ,but version 1.1 of this software supports Affymetrix 6.0 data files with additional data conversion steps. *PennCNV* [27] is the last software tool that we use for CNV detection for our comparative analyses. Although it is also designed to handle signal intensity data from Illumina arrays, it currently supports Affymetrix.

Comprehensive experimental results show that ÇOKGEN outperforms all of these four CNV identification tools in terms of general trio discordance. Overall, ÇOKGEN has a 30.8% discordance rate whereas Birdseye, dChip, QuantiSNP and PennCNV demonstrate discordance rates of 42.6%, 94%, 74% and 32.9% respectively on the same array data. It is important to note that dChip was originally optimized for detecting somatic copy number aberrations in cancer cells from earlier versions of the Affymetrix platform, and QuantiSNP is designed for data obtained from Illumina platform. Therefore, Birdseye, PennCNV, and ÇOKGEN's superior performance compared to dChip and QuantiSNP on Affymetrix 6.0 data is not surprising. For this reason, we restrict our assessment to ÇOKGEN, Birdseye and PennCNV in the remainder of this section.

As discussed in the previous section, the expected discordance rate of any algorithm approaches zero as it calls the CNV in more samples. 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 4, we first examine how the discordance rate behaves across call frequency strata for ÇOKGEN, PennCNV, and Birdseye. As a reference, we also display the expected discordance of randomly called CNVs in this figure. As expected, the performance of all algorithms improves when more frequent CNVs are considered. Although the performance of PennCNV is similar to that of ÇOKGEN, our algorithm does attain a modest improvement in concordance over PennCNV at all strata. It is also clear in Figure 4 that ÇOKGEN outperforms Birdseye significantly at all strata. Furthermore, 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 4 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. Figure 5A shows the empirical density for sample frequency of concordant CNVs. We find that 34% of the concordant CNVs identified by Birdseye have frequency larger than 60, whereas only 16% of the concordant CNVs identified by our algorithm and 14% of the CNVs identified by PennCNV have frequency larger than 60. Concordant CNVs with sample frequency larger than 90 make up 3% of those called

by our algorithm and 4% of those called by PennCNV as compared to 22% for Birdseye. This clearly shows that ÇOKGEN does not achieve its high concordance rate by overcalling a CNV in multiple samples. Figure 5B displays the density distribution of discordant CNVs as a function sample frequency for all algorithms. It is clear from the figure 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.

Sensitivity comparison across methods

Trio discordance is a reasonable hybrid measure of sensitivity (recall) and specificity (precision), but these two measures cannot be easily decoupled based only on discordance rate. A recent study [28] assembled a "stringent dataset", comprising CNVs identified by at least two independent algorithms. The dataset contains a total of 808 autosomal CNV regions reported by the study to be harbored in at least one of the 270 HapMap individuals. Another study [23] identified 1292 autosomal copy number polymorphism (CNP) regions in 270 HapMap samples. We use these two as "gold standard" data sets in which to evaluate the sensitivity of our method. We refer to sensitivity based on the data presented in [28] as *sensitivity-Pinto* and sensitivity based on the CNP data set presented in [23] as *sensitivity-McCarroll*.

In terms of *sensitivity-Pinto*, we observe that ÇOKGEN detects 696 of 808 (≈86%) CNVs from the study presented in [28]. PennCNV obtains the best result by a narrow margin, by identifying 716 of 808 (≈88.6%) CNVs. Birdseye achieves an 84.7% success rate,

slightly less than that of our method. In terms of *sensitivity-McCarroll*, ÇOKGEN, and PennCNV detect 20.7% and 25.5%, respectively. Birdseye detects 68.2%, which is the best sensitivity rate among all the methods compared for this data set; however, as mentioned in [23], Birdeye is one of the methods used for identifying the CNPs in this dataset. For this reason, this result is not surprising. PennCNV is slightly more sensitive than our method on this dataset, though this seems to be at the cost of a modest increase in trio discordance rate, as shown above.

Run Time Performance

To analyze the run time performances of ÇOKGEN, PennCNV, and Birdseye,we compare ÇOKGEN with PennCNV on a Windows system, and time both ÇOKGEN and Birdseye on a Linux system (Birdseye is not available in a Windows version). Performances are measured from the time at which the CEL file is taken as an input to the time at which the list of CNVs is output. On a Windows system that has an Intel Core 2 Quad CPU with a clock speed of 2.4 GHz and 4 gigabytes of memory, we observe that ÇOKGEN processes 22 chromosomes of a single HapMap sample in an average of 343 seconds compared to an average of 271 seconds for PennCNV package.

The Linux experiments are done on a Dual Intel Xeon 3Ghz Centos 5 x86 64 bit machine with 4 gigabytes of memory. Since Birdsuite is designed to be run as a pipeline of consecutive steps, we are unable to run only the Birdseye in isolation. Thus, we report the run time for the whole package rather than single steps, which may admittedly inflate the time that Birdseye would take to run alone. In this experiment, ÇOKGEN processes 22

chromosomes of a single sample in an average of 702 seconds compared to 2232 seconds for the whole Birdsuite software.

In addition to computational efficiency, these experiments also highlight the user-friendliness of our package. Indeed, ÇOKGEN is wholly contained in a single, simple (comprised of three commands) R package, making it completely platform-independent and available to Windows, Mac, or Linux/UNIX users. In contrast to the competing software, ÇOKGEN does not require the installation of additional tools such as Active Perl [29] or Affymetrix Power Tools [30].

Experimental Validation of CNVs not Previously Reported

To gauge the ability of ÇOKGEN to uncover novel gains and losses, we compared the CNVs discovered by our method with those in the version 6 (November 2008) of Database of Genomic Variants (DGV) [31]. We used multiplex ligation-dependent probe amplification (MLPA) [32] to verify some of the CNVs which are not reported in the DGV but are identified by ÇOKGEN, as shown in Table 2. In Figure 6, we also present the raw copy signal graphs generated by our software and the corresponding MLPA profiles for the first two CNVs given in Table 2.

The Software Package

Our software package, ÇOKGEN, which is implemented in R, 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 Figures 6 and 7). 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.

Conclusions

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, and the 6.0 array has the highest genome-wide resolution of current commercially-available platforms. 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 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.

The method described in this paper could also be adapted to other SNP arrays, including earlier versions of the Affymetrix platform, Illumina arrays, or array comparative genomic hybridization. Any platform that produces a measure of raw copy number at markers across the genome would be suitable. As SNP arrays continue to improve with regard to throughput and accuracy, our approach will be adaptable to handle the data as it becomes available.

The optimization based approach is the key to our method's flexibility. Although we have constructed our own default function to capture the criteria that we wish to emphasize, one may easily envision alternate criteria that other researchers would wish to incorporate. For example, since very long CNVs are quite rare in the human genome, researchers might wish to include a term in the objective function that takes into account the number of bases covered by a putative CNV region. Another possibility would be to incorporate allelic ratio intensity information at SNP markers, as is done in some hidden Markov model approaches [26,27]. We anticipate that users will design their own objective functions and apply them, using our software, to their own specific applications and data.

It should be emphasized that previously established approaches may actually also be considered special cases of functional optimization. For example, hidden Markov models

(HMMs) often used in the copy number setting [14, 17, 26] entail finding "state paths" (marker-by-marker sequences of copy-number calls) that maximize a log-likelihood function. In HMM applications, however, the model parameters are often estimated simultaneously with the copy number states via a Viterbi algorithm [33], based on training samples. Precise parameter estimation relies on sufficient representation from each copy number state, which may be unrealistic for rare CNVs. Another popular approach to inferring CNVs from raw copy number data is circular binary segmentation (CBS) [19]. Rather than explicitly representing copy number state as a solution to an optimization problem, CBS aims to find change points from one copy state to another. It does so by maximizing functions of marker indices. The optimum values of the function determine the boundaries of the CNV regions. A third example is the GLAD (Gain and Loss Analysis of DNA) algorithm [22], which has been adapted extensively by methods developed to analyze tumor DNA [15, 34]. To find CNVs, GLAD explicitly models raw copy number as a function of position. The true underlying copy number is encoded in a position-dependent parameter. The CNV regions are inferred by maximizing a weighted likelihood function using an adaptive weights smoothing procedure [21]. Note that the objective functions in HMMs, binary segmentation and GLAD all make distributional assumptions about the raw copy number measurements. The function that we adopt in the current study makes no such assumptions, but could be modified to incorporate such assumptions. Furthermore, our CNV calling method is fully unsupervised in that it does not require any training samples in terms of known copy numbers. Last, rather than estimating and fixing parameters (thus fixing the performance of the algorithm), our

method presents the opportunity to tune parameters, which makes it possible to adjust the performance of the algorithm to obtain the best results in a semi-automatic manner.

Three other studies utilize various smoothing and edge detection algorithms: wavelet footprints [35], non-linear diffusion filtering [36], and kernel smoothing [37]. We also apply an edge detection scheme on low-pass filtered data to identify regions that potentially correspond to aberrations. Unlike other approaches, however, we apply edge detection rather aggressively to identify all candidate regions that may correspond to aberrations. This is because the raw copy number signal is extremely noisy due to the artifacts of microarray technology, as seen in Figure 7A. Furthermore, since the markers are distributed unevenly across the genome, the one-dimensional signal represents a non-uniform sample of the actual copy number signal. Consequently, it is not straightforward to choose a smoothing and edge detection scheme that will be most appropriate for all experiments, samples, chromosomes, or even chromosomal segments. For example, in Figure 7B, the edge detection scheme identifies a single duplication as two separate duplications, since the markers at the middle of the region exhibit relatively low raw copy numbers, probably due to noise. This problem can be alleviated by smoothing the signal more aggressively to eliminate such artifacts; however this might result in falsely eliminating many aberrations that span relatively less numbers of markers. Motivated by these considerations, we use edge detection to identify all potential candidates and then use an optimization scheme with adjustable parameters to eliminate false calls among these candidates.

We also note that ÇOKGEN works on each sample individually and therefore is suited for rare CNV identification at the expense of losing some information to detect CNPs. The importance of rare CNVs is underscored by the recent deep sequencing of the entire genome of a single individual [38]. In that study, some 30% of the discovered CNVs had not been previously reported by any other study.

In addition to presenting a new software tool, the current work also casts Mendelian concordance, as an assessment tool, in a new light. While concordance rate is valuable as a metric to evaluate methods for calling germline variation, it is best viewed as a function of overall variant call rate. As we have shown, concordance rate can be artificially boosted simply by calling variants at a high rate. When evaluating the performance of future methods on family-based data sets, researchers may compare trio discordance results as a function of call frequency to the null expectation that we derive in the Materials and Methods section.

Materials and methods

Our method takes as input the raw .CEL files, and produces a table of inferred gains and losses, genome-wide. The software package, ÇOKGEN, 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. An overview of

the methods implemented in ÇOKGEN is given in Figure 1. Details of each step are provided in this section.

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* [39] to extract the intensities for each array locus from .CEL files. Next, we quantile normalize [13] 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.

Raw Copy Number for SNP 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

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

via least-squares regression, where $Z_i^{(A)}$ is the rescaled 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. 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 [23]. 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.

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

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

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

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: $R_i = \sqrt{A_i^2 + B_i^2}$.

Raw Copy Numbers for CN Markers

Approximately half of the marker loci represented on the 6.0 array do not correspond to SNPs, but rather 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

$$2 = \beta_i PI_i + e_i ,$$

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, ordered along the genome according to hg18 (build 36 of the human genome) coordinates. All 270 HapMap samples are used to parameterize the regression model for raw copy number estimation of both SNP and CNV markers. Figure 7A gives an example of raw copy numbers for a 394-marker region.

Algorithm for Copy Number Variant Detection

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: $\{1,..., N\} \rightarrow \mathcal{C}$, where $\{1,..., N\}$ denotes the ordered set of markers for the whole genome and $\mathcal{C} = \{C_+, C_0, C_-\}$ 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 $\mathcal{C}$ 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.

We next 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.

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 $\Pi(c) = \{i \in \{1,...,M\} : S(i) = c\}$ 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

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

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.

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).

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).

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 2.25 and 1.75 for T_{gain} and T_{loss} respectively, since these values provide reasonable performance.

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^*: $\{1,\ldots, N\} \rightarrow \mathcal{C} = \{C_+, C_0, C_-\}$ such that the function

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

is minimized at $S = S^*$. Here the tunable coefficients $k_\sigma, k_\chi, k_\lambda, k_\delta$ adjust the relative importance of the objective criteria with respect to each other. 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 conduct a series of computational experiments. The results of these experiments are presented in Figure 8. Here, sensitivity is a measure of the performance of the algorithm in capturing previously reported CNVs. As seen in the figure, *sensitivity-Pinto* rapidly improves as more weight is assigned to variability and nears saturation around $k_\sigma = 0.35$ and $k_\chi = 0.65$. On the other hand, *sensitivity-McCarroll* keeps improving until it settles around $k_\sigma = 0.6$ and $k_\chi = 0.4$. In contrast, discordance rapidly declines as we increase the contribution of variability to the objective function, achieves a minimum around $k_\sigma = 0.35$ and $k_\chi = 0.65$, and grows until it settles around 0.8 for $k_\sigma = 0.8$ and $k_\chi = 0.2$. 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.

Two Phase CNV Identification

Since the solution space of the optimal copy number assignment problem is exponential, 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 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 7B demonstrates how the raw copy numbers R_i in Figure 7A is converted into a smooth signal R_i^* using the low pass filter.

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 $\hat{R}:[0,\ \mathrm{M}]\rightarrow\Re$. 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

$$D_{\max} = \{i \in 2,...,\mathrm{M} \mid \hat{R}'(i) > \hat{R}'(i-1) \text{ and } \hat{R}'(i) > \hat{R}'(i+1)\}$$
$$D_{\min} = \{i \in 2,...,\mathrm{M} \mid \hat{R}'(i) < \hat{R}'(i-1) \text{ and } \hat{R}'(i) < \hat{R}'(i+1)\}$$

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

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 (iv) ensures that Q_{ij} can be enlarged neither at the right nor the left borders. Examples of regions that violate each of the conditions are shown in Figure 9. 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 $S^{(0)}$. This 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.

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. 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 10. As seen in the figure, we restrict possible moves to those that can enlarge or shrink a candidate aberrant region, but can never create a candidate region from scratch or divide a candidate region into two candidate regions. The size of the valid moves set that can shrink a candidate region Q_{ij} of size n is $2n$-1. This set contains all moves that change the class value of contiguous markers in Q_{ij}'s either left or right boundaries to the class of neighboring markers outside Q_{ij}. The size of the valid moves that can enlarge a candidate region Q_{ij} is limited to 2ψ (i.e., at most ψ markers from left and ψ from right are converted to the class of region Q_{ij}) where ψ is a user defined parameter which determines how aggressively a candidate region to be enlarged. In our experiments, we set $\psi = 5$ that limits the permissible expansion of a CNV region. We set such a threshold since we want our algorithm to expand the candidate region gradually. Thus, the total number of valid moves at each stage of our algorithm for a candidate region of size n is $2n + 9$.

We quantify the quality of such 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. In the context of the optimal copy number assignment problem, 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. We use a stochastic algorithm that is based on simulated annealing [40] to determine the move. 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. The procedure is repeated until either there is no valid 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^*.

We note that our algorithm allows us to consider the candidate regions in $\varsigma(S)$ independently (as opposed to the entire chromosome) because the candidate regions with potential aberrations are sparse, and therefore we work on local sub-problems associated with each candidate region separately. This results in significant improvement in

computational efficiency. Since the distribution of raw copy numbers in the neighborhood of $Q_{ij} \in \varsigma(S)$ provides a good sampling of raw copy numbers in the entire chromosome, the quality of the solution to this local problem does not deviate significantly from the global problem. Indeed, in our experimental evaluations, we observe that there is no significant difference between solving the class assignment globally (applying the above algorithm to whole chromosome) or locally (as we describe above) in terms of their specificity and sensitivity in predicting copy number variations.

Data

For the application of our method, we used Affymetrix 6.0 array data from a total of 270 HapMap individuals. In the data set, there are 30 mother-father-child trios from the Yoruba people of Ibadan, Nigeria, 45 unrelated individuals from the Tokyo, Japan, 45 unrelated individuals from Beijing, China and thirty Caucasian trios that were collected in 1980 from U.S. residents with northern and western European ancestry by the Centre d'Etude du Polymorphisme Humain (CEPH).

MLPA Method

Each MLPA probe was designed synthetically to match sequences within the region of interest avoiding all SNPs in the area. Control probes were used from previously published work [41]. Oligonucleotides were synthesized by IDT, Coralville, IA, with 5'-phosphorylation of each downstream probe and tagged with common PCR primer sequences [32]. Probes were hybridized with 100 ng of DNA sample using MLPA reagents (part number EK1, MRC-Holland BV, Amsterdam, The Netherlands) in accordance with the recommended protocol. Samples were diluted 20 fold and analyzed

on a 3130xl Genetic Analyzer from Applied Biosystems with GeneMapper software. Control DNA used were male and female genomics DNA pools (Promega, Madison, WI). Peak heights ratios were normalized to the mean of the entire data set, with subsequent elimination of outlier samples from the calculation of the mean.

Other Methods

For analysis using dChip [24], we downloaded the version with a build date of August 21, 2008. We used the hidden Markov model as the Inferred copy method option with 50% of the samples trimmed. For Birdseye [17], we used version 1.5.1 of the Birdsuite package, which can be downloaded from [42]. The default parameters for that package were used. The latest version of PennCNV software which was available as of November 18, 2008 was downloaded from [43] for analysis using PennCNV. We followed the steps described at [44] for PennCNV-Affy protocol and used the default parameters for analysis. For QuantiSNP, we downloaded version 1.1 from [45], followed the steps described at QuantiSNP for Affymetrix tutorial document located at [46] and used the default parameters.

We also note that we combined copy number 0 and 1 into one category – loss – and copy number greater than 2 into one category – gain – for the results obtained by all packages, in order to compare their results with ÇOKGEN's results fairly.

Computation of Expected Discordance Rate

Suppose that CNV calls are random in Φ parent-child trios. When randomly assigning a CNV to n of the 3Φ individuals, the expected discordance, denoted by $ED(n)$, can be calculated as:

$$ED(n) = \sum_{k=1}^{\min(\Phi,n)} ED(n \mid k) \times P(k)$$

where $ED(n \mid k)$ denotes the expected discordance rate when k of the n CNVs are assigned to children and $P(k)$ denotes the probability of assigning the CNV to k children. We calculate $P(k)$ as

$$P(k) = \frac{\binom{\Phi}{k} \times \binom{2\Phi}{n-k}}{\binom{3\Phi}{n}} .$$

$ED(n \mid k)$ is equivalent to the probability of having a child assigned a CNV being discordant given that k children have the CNV. This probability can be calculated by dividing the number of ways to assign this CNV to parents other than the discordant child's parents by the number of all possible ways to assign this CNV to parents. Thus, $ED(n \mid k)$ is

$$ED(n \mid k) = \frac{\binom{2\Phi - 2}{n-k}}{\binom{2\Phi}{\min(2\Phi, n-k)}}.$$

Consequently,

$$ED(n) = \sum_{k=1}^{\min(\Phi,n)} \left[\frac{\binom{2\Phi - 2}{n-k}}{\binom{2\Phi}{\min(2\Phi, n-k)}} \times \frac{\binom{\Phi}{k} \times \binom{2\Phi}{n-k}}{\binom{3\Phi}{n}} \right].$$

Abbreviations

ÇOKGEN (chok-gen); An amalgamation of the Turkish words ÇOK = multiple and GEN = gene, SNP; Single Nucleotide Polymorphism, CNV; Copy Number Variation, CNP; Copy Number Polymorphism, HMM; Hidden Markov Model, DGV; Database of Genomic Variants, MLPA; Multiplex Ligation-Dependent Probe Amplification

Authors' contributions

GY, MK and TL designed the algorithms. MÖ provided constructive discussions for the development of the algorithm. GY implemented the ÇOKGEN framework and collected the results for analysis. GY, MK and TL developed methodology for *in silico* analysis of the results and GY analyzed the results. MG did the MLPA validation of the not previously reported CNVs. All authors read and approved the final manuscript.

Acknowledgements

The authors express thanks to Sivakumaran Arumugam and Katherine Wilkins for running the Birdseye and dChip software, respectively. We also thank Simone Edelheit and the CWRU Genomics Core Facility for assistance with MLPA analysis. This work was supported by National Cancer Institute Grant R01 CA131341 to TL.

References

1. International HapMap Consortium: **A haplotype map of the human genome**. *Nature* 2005, **437**:1241-2.

2. Affymetrix: **Genome-Wide Human SNP Array 6.0 data sheet**. 2007, Santa Clara (California).

3. Illumina: **Human1M-duo beadchip data sheet**. 2007, San Diego (California).

4. Feuk L, Carson AR, Scherer SW: **Structural variation in the human genome**. *Nat Rev Genet* 2006, **7**:85-97.

5. Rovelet-Lecrux A, Hannequin D, Raux G, Le Meur N, Laquerrière A, Vital A, Dumanchin C, Feuillette S, Brice A, Vercelletto M, Dubas F, Frebourg T, Campion D: **APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy**. *Nat Genet* 2006, **38**:24-26.

6. Fellermann K, Stange DE, Schaeffeler E, Schmalzl H, Wehkamp J, Bevins CL, Reinisch W, Teml A, Schwab M, Lichter P, Radlwimmer B, Stange EF. **A chromosome 8 gene-cluster polymorphism with low human beta-defensin 2 gene copy number predisposes to Crohn disease of the colon**. *Am J Hum Genet* 2006, **79**:439-448.

7. Sebat J, Lakshmi B, Malhotra D, Troge J, Lese-Martin C, Walsh T, Yamrom B, Yoon S, Krasnitz A, Kendall J, Leotta A, Pai D, Zhang R, Lee YH, Hicks J, Spence SJ, Lee AT, Puura K, Lehtimäki T, Ledbetter D, Gregersen PK, Bregman J, Sutcliffe JS, Jobanputra V, Chung W, Warburton D, King MC, Skuse D, Geschwind DH, Gilliam TC, et al.: **Strong association of de novo copy number mutations with autism**. *Science* 2007, **316**:445-449.

8. Xu B, Roos JL, Levy S, van Rensburg EJ, Gogos JA, Karayiorgou M: **Strong association of de novo copy number mutations with sporadic schizophrenia**. *Nat Genet* 2008, **40**:880-885.

9. Zhao X, Li C, Paez JG, Chin K, Jänne PA, Chen TH, Girard L, Minna J, Christiani D, Leo C, Gray JW, Sellers WR, Meyerson M: **An integrated view of copy number and allelic alterations in the cancer genome using single nucleotide polymorphism arrays**. *Cancer Res* 2004, **64**:3060-3071.

10. Peiffer DA, Le JM, Steemers FJ, Chang W, Jenniges T, Garcia F, Haden K, Li J, Shaw CA, Belmont J, Cheung SW, Shen RM, Barker DL, Gunderson KL: **High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping**. *Genome Res* 2006, **16**:1136-1148.

11. Gunderson KL, Steemers FJ, Lee G, Mendoza LG, Chee MS: **A genome-wide scalable SNP genotyping assay using microarray technology**. *Nat Genet* 2005, **37**:549-554.

12. Lindblad-Toh K, Tanenbaum DM, Daly MJ, Winchester E, Lui WO, Villapakkam A, Stanton SE, Larsson C, Hudson TJ, Johnson BE, Lander ES, Meyerson M: **Loss-of-heterozygosity analysis of small-cell lung carcinomas using single-nucleotide polymorphism arrays**. *Nat Biotechnol* 2000, **18**:1001-1005.

13. Bolstad BM, Irizarry RA, Astrand M, Speed TP: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias**. *Bioinformatics* 2003, **19**:185-193.

14. Lin M, Wei LJ, Sellers WR, Lieberfarb M, Wong WH, Li C: **dChipSNP: significance curve and clustering of SNP-array-based loss-of-heterozygosity data**. *Bioinformatics* 2004, **20**:1233-1240.

15. LaFramboise T, Harrington D, Weir BA: **PLASQ: a generalized linear model-based procedure to determine allelic dosage in cancer cells from SNP array data**. *Biostatistics* 2007, **8**:323-336.

16. Bengtsson H, Irizarry R, Carvalho B, Speed TP: **Estimation and assessment of raw copy numbers at the single locus level**. *Bioinformatics* 2008, **24**:759-767.

17. Korn JM, Kuruvilla FG, McCarroll SA, Wysoker A, Nemesh J, Cawley S, Hubbell E, Veitch J, Collins PJ, Darvishi K, Lee C, Nizzari MM, Gabriel SB, Purcell S, Daly MJ, Altshuler D: **Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs**. *Nat Genet* 2008, **40**:1253-1260.

18. Zhao X, Weir BA, LaFramboise T, Lin M, Beroukhim R, Garraway L, Beheshti J, Lee JC, Naoki K, Richards WG, Sugarbaker D, Chen F, Rubin MA, Jänne PA, Girard L, Minna J, Christiani D, Li C, Sellers WR, Meyerson M: **Homozygous deletions and chromosome amplifications in human lung carcinomas revealed by single nucleotide polymorphism array analysis**. *Cancer Res* 2005, **65**:5561-5570.

19. Olshen AB, Venkatraman ES, Lucito R, Wigler M: **Circular binary segmentation for the analysis of array-based DNA copy number data**. *Biostatistics* 2004, **5**:557-572.

20. Venkatraman ES, Olshen AB: **A faster circular binary segmentation algorithm for the analysis of array CGH data**. *Bioinformatics* 2007, **23**:657-663.

21. Polzehl J, Spokoiny S: **Adaptive weights smoothing with applications to image restoration**. *J R Stat Soc, Ser B* 2000, **62**:335–354.

22. Hupé P, Stransky N, Thiery JP, Radvanyi F, Barillot E: **Analysis of array CGH data: from signal ratio to gain and loss of DNA regions**. *Bioinformatics* 2004, **20**:3413-3422.

23. McCarroll SA, Kuruvilla FG, Korn JM, Cawley S, Nemesh J, Wysoker A, Shapero MH, de Bakker PI, Maller JB, Kirby A, Elliott AL, Parkin M, Hubbell E, Webster T, Mei R, Veitch J, Collins PJ, Handsaker R, Lincoln S, Nizzari M, Blume J, Jones KW, Rava R, Daly MJ, Gabriel SB, Altshuler D: **Integrated detection and population-genetic analysis of SNPs and copy number variation**. *Nat Genet* 2008, **40**:1166-1174.

24. **dChip Software Website** [http://www.dchip.org].

25. Li C, Wong WH: **Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection**. *PNAS* 2001, **98**:31-36.

26. Colella S, Yau C, Taylor JM, Mirza G, Butler H, Clouston P, Bassett AS, Seller A, Holmes CC, Ragoussis J: **QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data**. *Nucleic Acids Res* 2007, **35**:2013-2025.

27. Wang K, Li M, Hadley D, Liu R, Glessner J, Grant SF, Hakonarson H, Bucan M: **PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data**. *Genome Res* 2007, **17**:1665-1674.

28. Pinto D, Marshall C, Feuk L, Scherer SW: **Copy-number variation in control population cohorts**. *Hum Mol Genet* 2007, **16**:R168-173.

29. **Active Perl** [http://www.activestate.com/activeperl/].

30. **Affymetrix Power Tools Software Package** [http://www.affymetrix.com/partners_programs/programs/developer/tools/powertools.affx].

31. **Database of Genomic Variants** [http://projects.tcag.ca/variation/].

32. Schouten JP, McElgunn CJ, Waaijer R, Zwijnenburg D, Diepvens F, Pals G: **Relative quantification of 40 nucleic acid sequences by multiplex ligation-dependent probe amplification**. *Nucleic Acids Res* 2002, **30**:e57.

33. Dempster AP, Laird NM, Rubin DB: **Maximum likelihood from incomplete data via the EM algorithm**. *Journal of the Royal Statistical Society* 1977, **39**:1-38.

34. Beroukhim R, Lin M, Park Y, Hao K, Zhao X, Garraway LA, Fox EA, Hochberg EP, Mellinghoff IK, Hofer MD, Descazeaud A, Rubin MA, Meyerson M, Wong WH, Sellers WR, Li C: **Inferring loss-of-heterozygosity from unpaired tumors**

using high-density oligonucleotide SNP arrays. *PLoS Comput Biol* 2006, **2**: e41.

35. Pique-Regi R, Tsau E-S, Ortega A, Seeger R, Asgharzadeh S: **Wavelet footprints and sparse bayesian learning for DNA copy number change analysis**. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 15-20 April 2007; Hawaii.* 2007, **1**:353-356.

36. Alqallaf A, Tewfik A, Selleck S, Johnson R: **Framework for the analysis of genetic variations across multiple DNA copy number samples**. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 30 March -4 April 2008; Las Vegas.* 2008, **1**:553-556.

37. Shen F, Huang J, Fitch KR, Truong VB, Kirby A, Chen W, Zhang J, Liu G, McCarroll SA, Jones KW, Shapero MH: **Improved detection of global copy number variation using high density, non-polymorphic oligonucleotide probes**. *BMC Genet* 2008, **9**:27.

38. Wang J, Wang W, Li R, Li Y, Tian G, Goodman L, Fan W, Zhang J, Li J, Zhang J, Guo Y, Feng B, Li H, Lu Y, Fang X, Liang H, Du Z, Li D, Zhao Y, Hu Y, Yang Z, Zheng H, Hellmann I, Inouye M, Pool J, Yi X, Zhao J, Duan J, Zhou Y, Qin J, et al.: **The diploid genome sequence of an Asian individual**. *Nature* 2008, **456**:60-65.

39. **Affymetrix File Parsing SDK** [http://www.bioconductor.org/packages/2.2/bioc/html/affxparser.html].

40. Kirkpatrick S, Gelatt CD Jr, Vecchi MP: **Optimization by Simulated Annealing**. *Science* 1983, **220**:671-680.

41. Macconaill LE, Aldred MA, Lu X, Laframboise T: **Toward accurate high-throughput SNP genotyping in the presence of inherited copy number variation**. *BMC Genomics* 2007, **8**:211.

42. **Birdsuite download page** [http://www.broad.mit.edu/science/programs/medical-and-population-genetics/birdsuite/birdsuite-downloads-0].

43. **PennCNV download page** [http://www.openbioinformatics.org/penncnv/penncnv_download.html].

44. **PennCNV-Affy tutorial page** [http://www.openbioinformatics.org/penncnv/penncnv_tutorial_affy_gw6.html].

45. **QuantiSNP download page** [http://www.well.ox.ac.uk/~ioannisr/quantisnp/]

46. **QuantiSNP for Affymetrix tutorial page** [http://groups.google.co.uk/group/quantisnp/files]

Figure Legends

Figure 1 - The overview of the proposed CNV detection algorithm

ÇOKGEN first extracts the intensity values from the Affymetrix .CEL files. It then obtains the raw copy numbers for each marker using regression with the help of the Affymetrix software's SNP genotype calls. The edge detection determines the candidate loss/gain regions from smoothed copy number signal, which is obtained by low-pass filtering the raw copy numbers. We determine the final class assignments using objective

function optimization. The function is optimized using an iterative simulated annealing procedure, with initialization provided by the edge detection.

Figure 2 - CNVs identified by ÇOKGEN

For each marker position on every chromosome, the gain or loss frequencies in the HapMap samples are plotted. The frequencies for gains are shown on the positive y-axis with green lines; and the loss frequencies are shown on the negative y-axis with blue lines.

Figure 3 - Frequency distribution of CNVs by ethnicity

The proportion of rarer CNVs (i.e., those that have a sample frequency less than 10) in the YRI population is higher when compared to the other populations.

Figure 4 - Discordance rate as a function of call frequency strata

The figure shows how the discordance rates behave as a function of the sample frequency threshold. Note that discordance rate is plotted cumulatively – that is, the value on the y-axis is the average discordance rate for CNVs with frequencies *at most* the corresponding value on the x-axis. The discordance value at the sample frequency threshold value *t* is calculated by finding the discordance rate across all CNVs with frequency at most *t*.

Figure 5 - The frequency distribution of (A) concordant CNVs and (B) discordant CNVs for three calling algorithms

(a) shows that ÇOKGEN's concordant CNVs are mostly rarer and **(b)** shows that its discordant CNVs are more frequent in the population, particularly when compared to those of Birdseye.

Figure 6 - MLPA profiles and corresponding raw copy signals with class assignments for two CNVs not previously reported

(a) and **(b)** show a representative gain and loss, respectively, with overlays of two traces from a multiplex ligation dependent probe amplification (MLPA). Red tracings represent pooled normal control sample with blue tracings showing the HapMap sample. Peaks not at or adjacent to the arrows represent control regions. The arrows indicate where the gain or loss occurs. **(c)** and **(d)** present the raw copy signals and ÇOKGEN's class assignments for the MLPA profiles in **(a)** and **(b)**, respectively. Here, ÇOKGEN inferences are colored red = normal, green = gain, blue = loss.

Figure 7 - Raw copy numbers for sample NA12763 in a chromosome 12 region

(a) The raw copy numbers R_i. **(b)** The smooth signal R_i^*, obtained by applying the 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. Note that there exist two candidate gain regions in the figure. **(c)** Our objective function optimization using simulated annealing makes the final assignments to the markers and it merges the two candidate regions in **(b)** into one gain region.

Figure 8 - The trio discordance and sensitivity as a function of k_σ

The figure shows how trio discordance and sensitivity are affected as we alter the relative weights of variability and parsimony in the objective function.

Figure 9 - Illustration of criteria for selection of candidate CNV regions

In this figure, the red lines tangent to the signal curve $\hat{R}(i)$ indicate the marker points that are elements of the sets $D_{\max}$ and $D_{\min}$. **(a)** Q_{ij} violates the first condition which

requires that the start and end markers be elements of D_{max} and D_{min} respectively. **(b)** Q_{ij} violates the second condition which dictates that at least one marker must exceed the T_{gain} threshold value. **(c)** Q_{ij} does not satisfy the condition 3 since three markers of D_{min} appear before two markers of D_{max} in region Q_{ij}. **(d)** Q_{ij} violates the last condition, which requires that Q_{ij} is a maximal set satisfying all the conditions (i), (ii) and (iii).

Figure 10 - Illustration of moves in the proposed iterative-improvement based optimization algorithm

The CNV assignment for a hypothetical region after t iterations is shown in the initial figure. Suppose Q_{ij} is selected for the $(t+1)^{st}$ iteration. **(a)** All the markers in Q_{ij} is assigned to C_0 which completely eliminates Q_{ij} as a gain region. This is a valid move. **(b)** Some markers which initially have a C_0 class on the right border of Q_{ij} are assigned to the C_+ class which merges them with Q_{ij} by a valid move. **(c)** Some markers which initially exist in Q_{ij} are assigned to the C_0 class which contracts Q_{ij}. This also represents a valid move. **(d)** This is an invalid move which divides the Q_{ij} into two sub gain regions. **(e)** This is another invalid move which introduces a completely new gain region that is not identified in the previous solution.

Table 1 - The distribution of identified CNVs by ethnicity.

	CEU	YRI	JPT	CHB	Total
Gains	1726	2325	856	765	5672
Losses	3500	3443	1760	1753	10456
Total	5226	5768	2616	2518	16128

Table 2 - MLPA results for some of the non-previously reported regions identified by ÇOKGEN

Chr	Sample	Bp Start*	Bp End*	Length (bp)	MLPA Probe Position	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
6	NA12144	99225007	99245596	20589	99226748	Loss	1.3
16	NA10839	77818007	77832838	14831	77819334	Loss	1.35
2	NA10854	108944933	108952869	7936	108945672	Loss	1.33
6	NA11830	97308635	97316868	8233	97311558	Loss	1.29

* As inferred by ÇOKGEN.

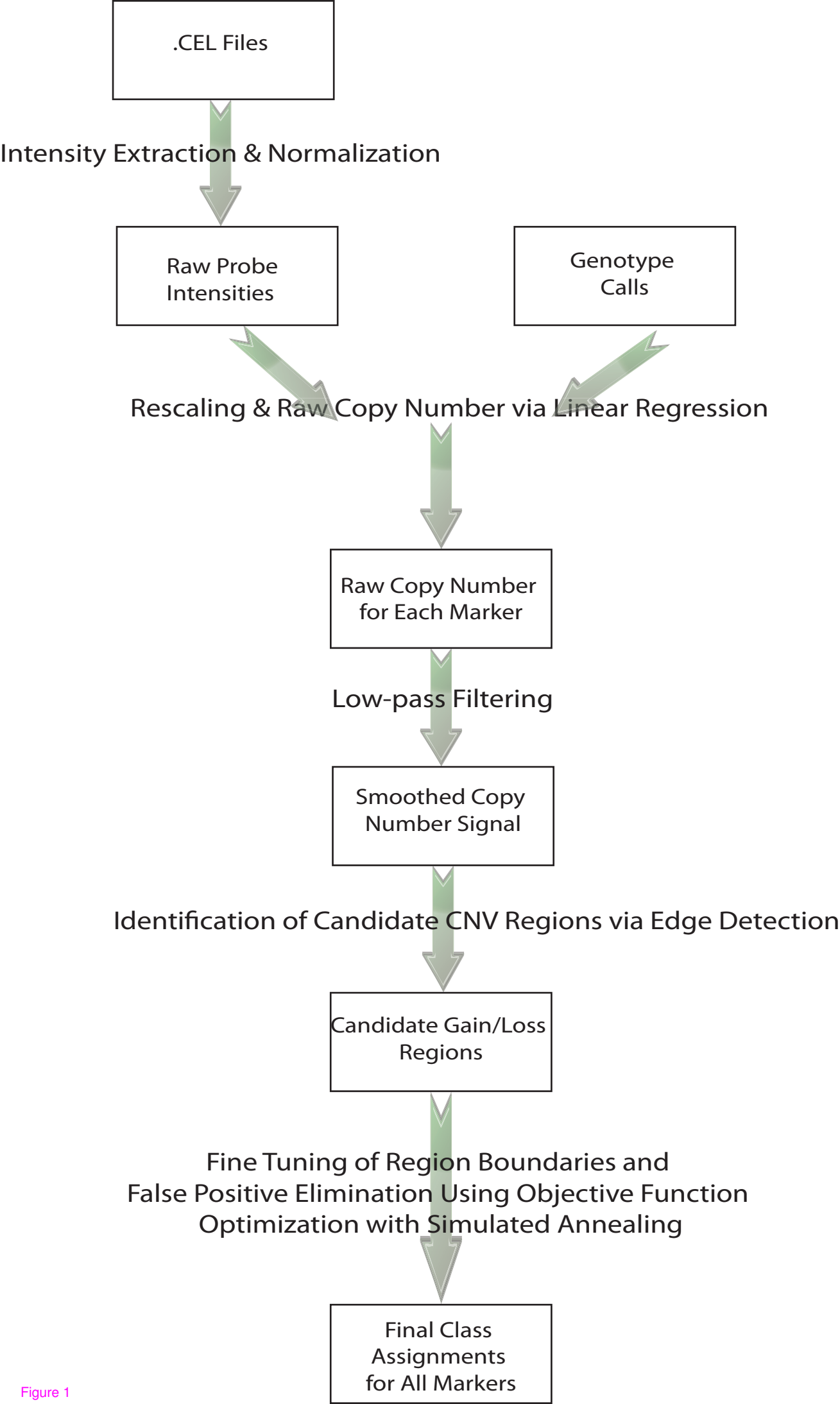

.CEL Files
Intensity Extraction & Normalization
Raw Probe Intensities
Genotype Calls
Rescaling & Raw Copy Number via Linear Regression
Raw Copy Number for Each Marker
Low-pass Filtering
Smoothed Copy Number Signal
Identification of Candidate CNV Regions via Edge Detection
Candidate Gain/Loss Regions
Fine Tuning of Region Boundaries and
False Positive Elimination Using Objective Function
Optimization with Simulated Annealing
Final Class Assignments for All Markers

Figure 1

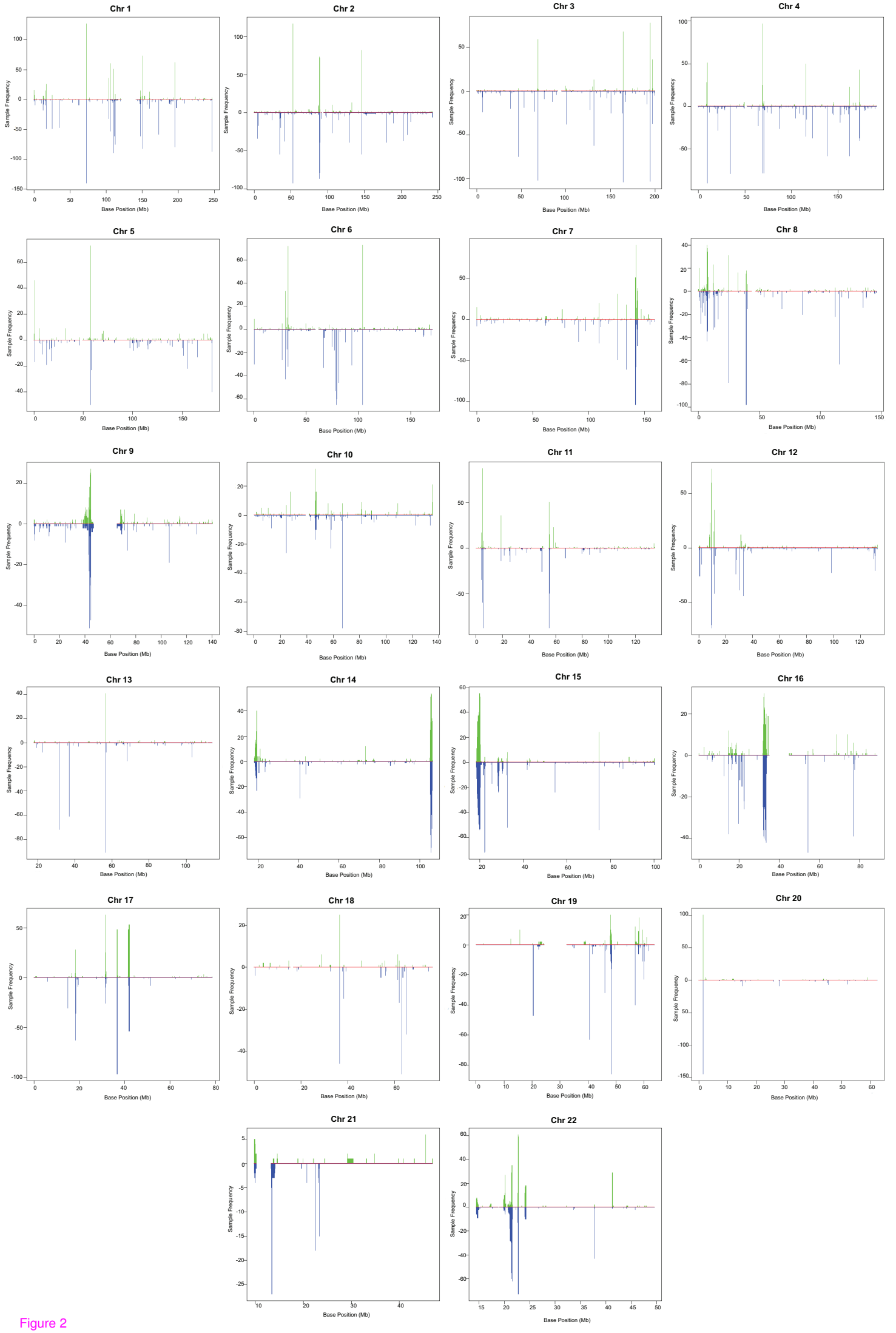

Chr 1
Chr 2
Chr 3
Chr 4
Chr 5
Chr 6
Chr 7
Chr 8
Chr 9
Chr 10
Chr 11
Chr 12
Chr 13
Chr 14
Chr 15
Chr 16
Chr 17
Chr 18
Chr 19
Chr 20
Chr 21
Chr 22
Sample Frequency
Base Position (Mb)

Figure 2

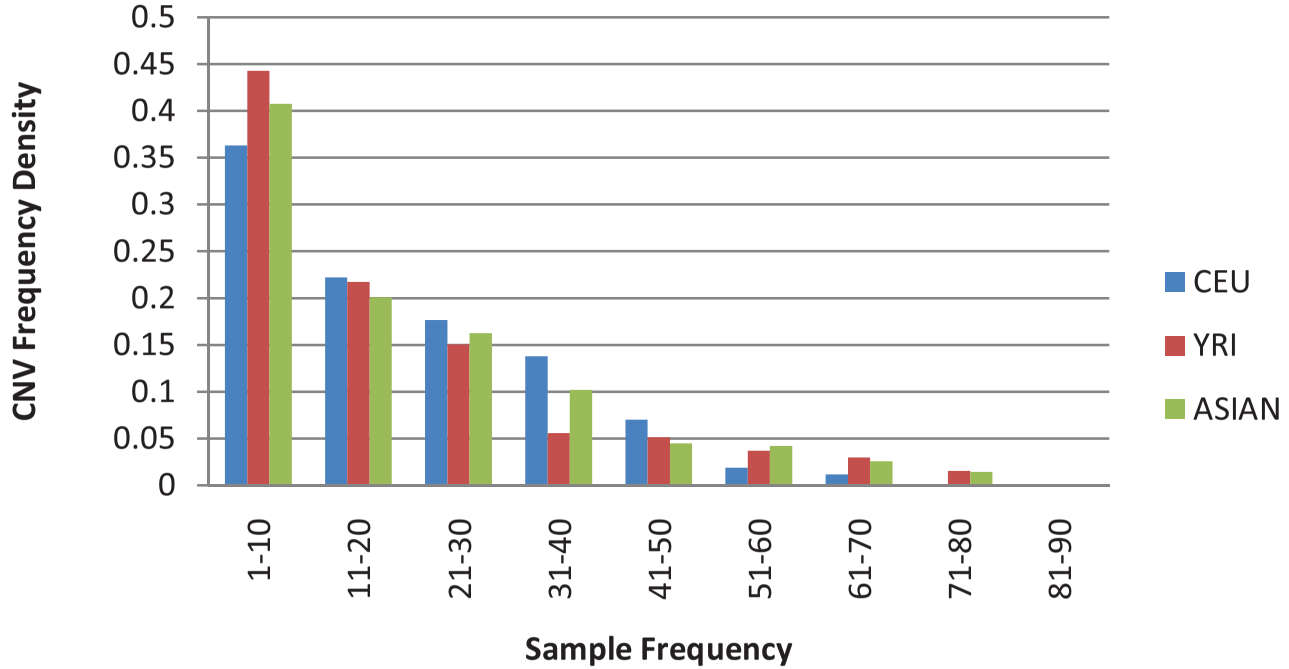

CNV Frequency Density
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1-10
11-20
21-30
31-40
41-50
51-60
61-70
71-80
81-90
Sample Frequency
CEU
YRI
ASIAN

Figure 3

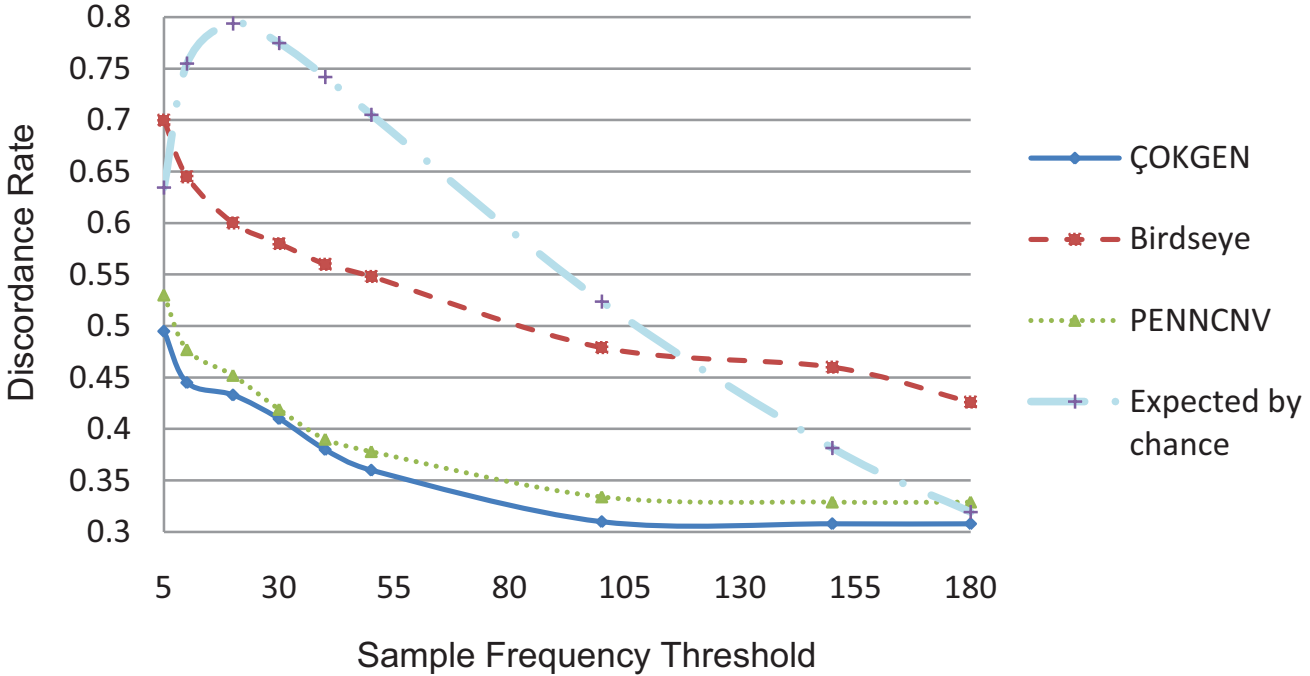

0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
Discordance Rate
5
30
55
80
105
130
155
180
Sample Frequency Threshold
ÇOKGEN
Birdseye
PENNCNV
Expected by chance

Figure 4

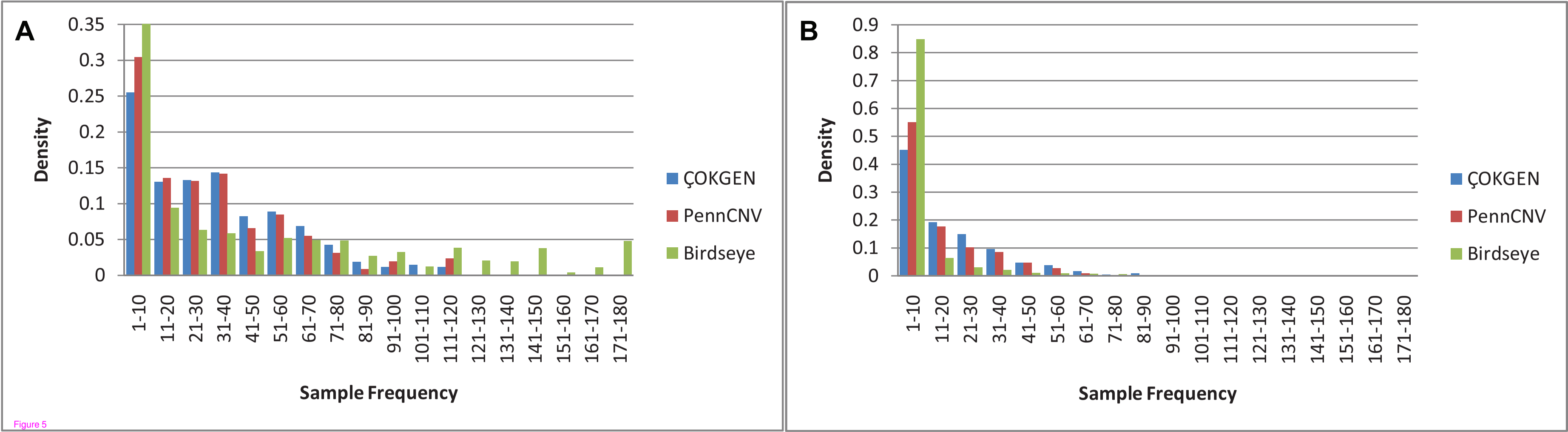
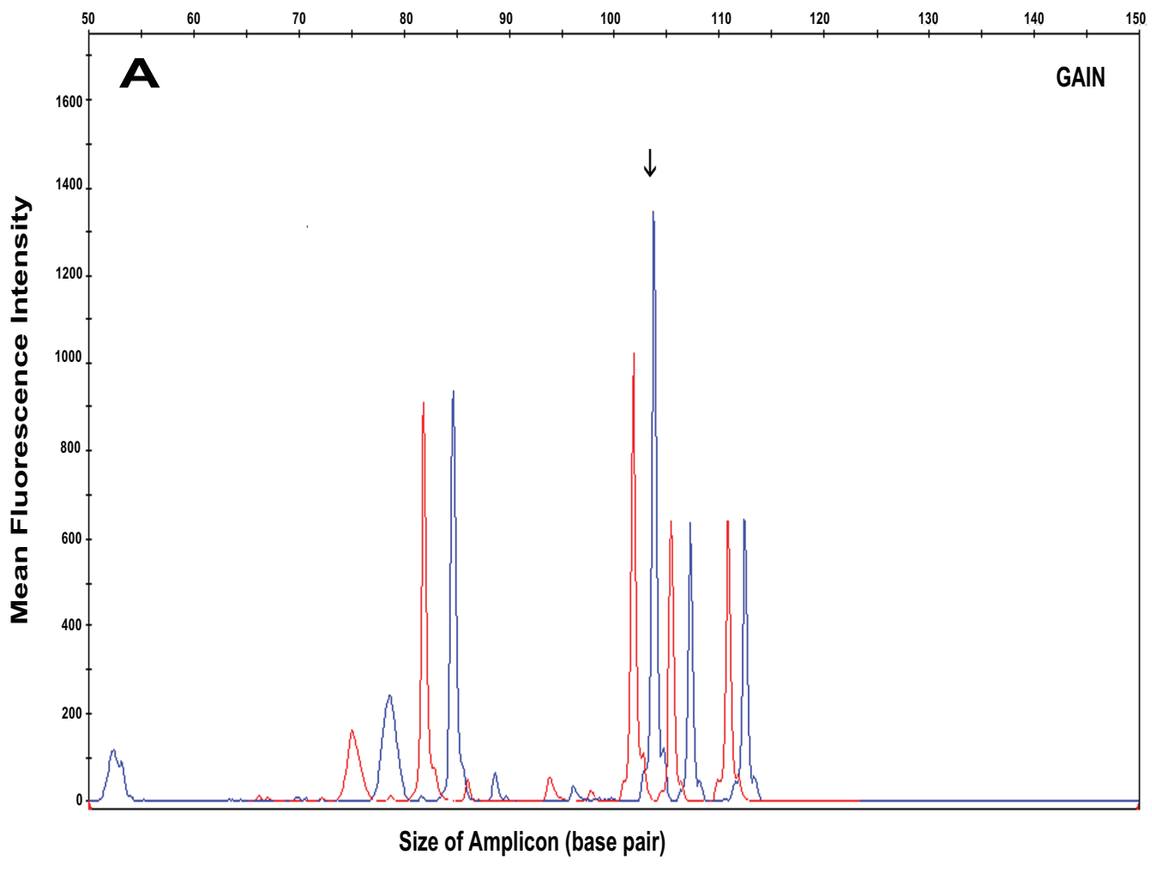
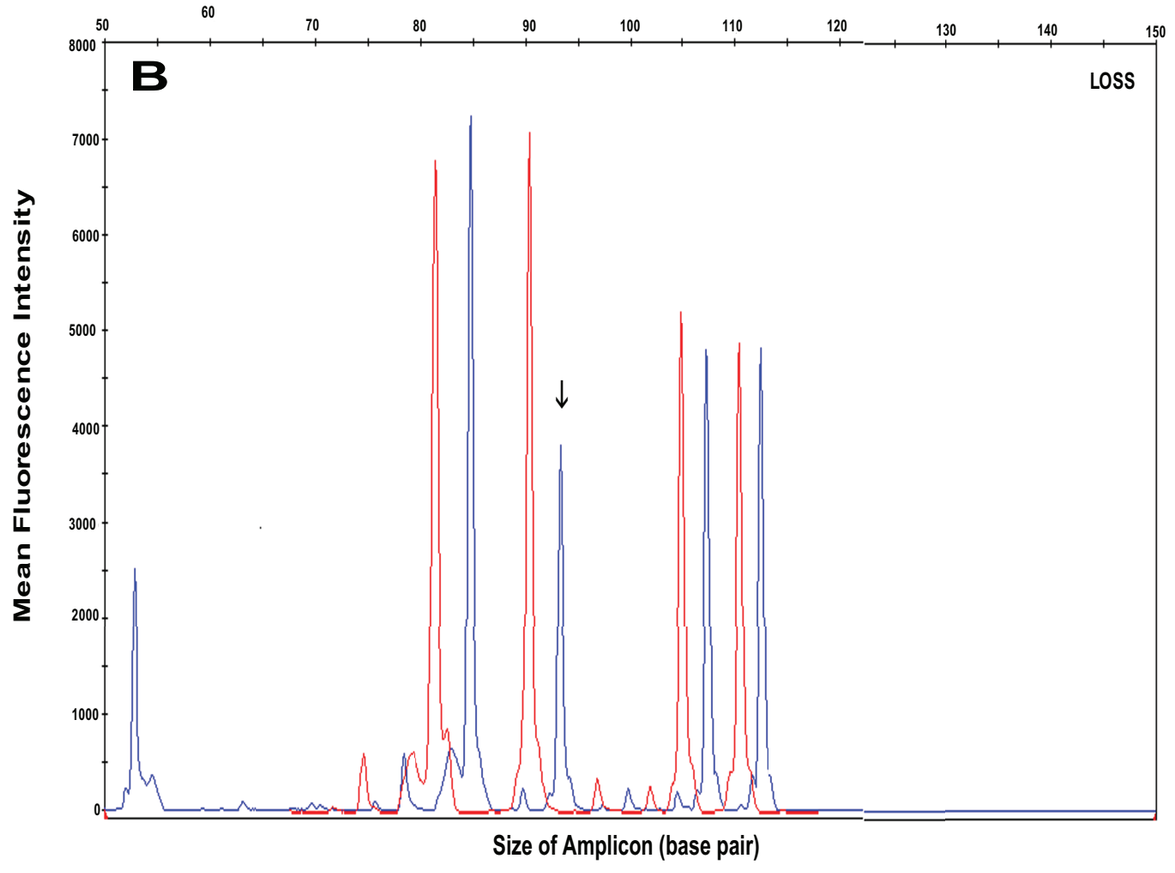
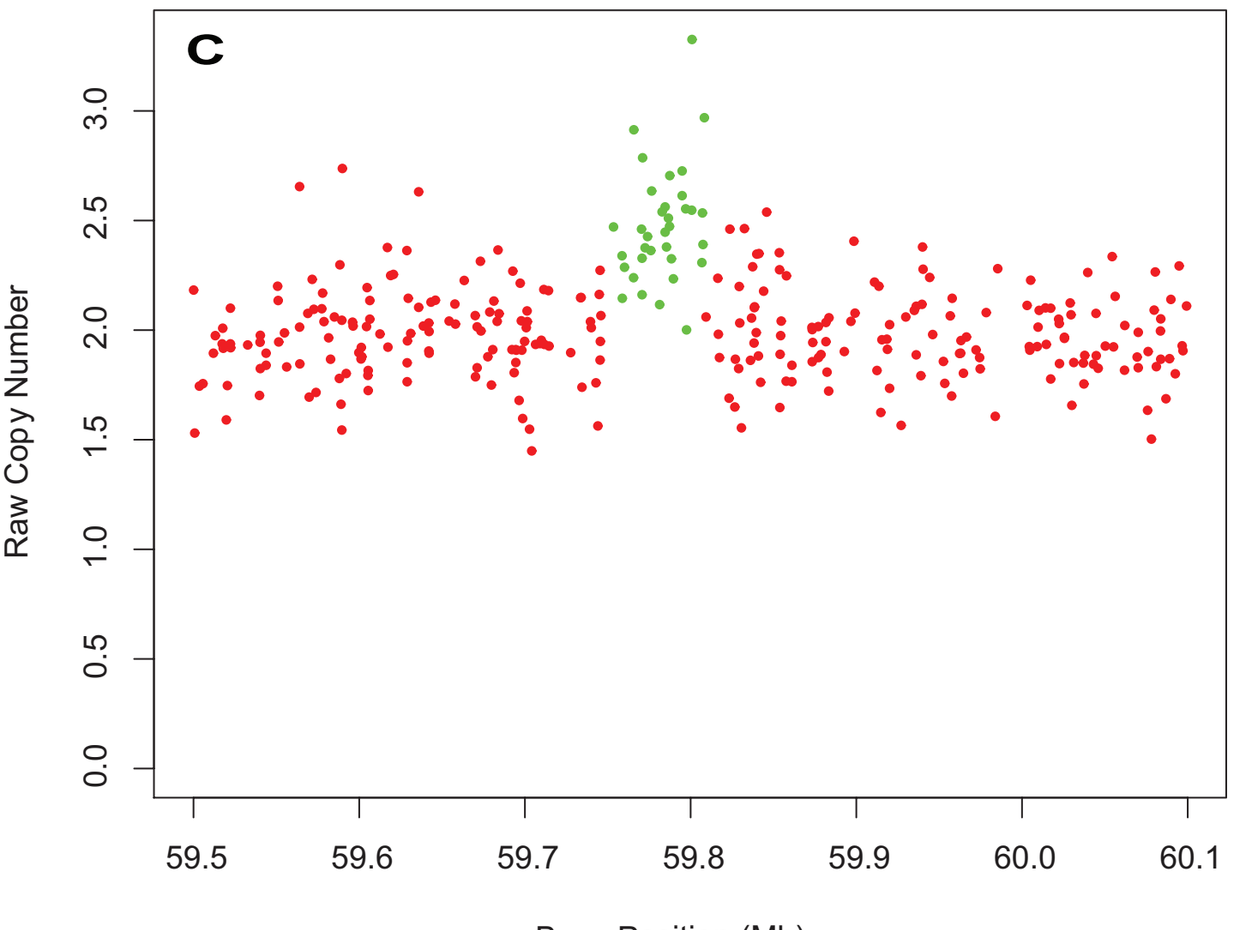

A
Density
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1-10
11-20
21-30
31-40
41-50
51-60
61-70
71-80
81-90
91-100
101-110
111-120
121-130
131-140
141-150
151-160
161-170
171-180
Sample Frequency
ÇOKGEN
PennCNV
Birdseye
B
Density
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1-10
11-20
21-30
31-40
41-50
51-60
61-70
71-80
81-90
91-100
101-110
111-120
121-130
131-140
141-150
151-160
161-170
171-180
Sample Frequency
ÇOKGEN
PennCNV
Birdseye

Figure 5


A
GAIN
Mean Fluorescence Intensity
Size of Amplicon (base pair)


B
LOSS
Mean Fluorescence Intensity
Size of Amplicon (base pair)


C
Raw Copy Number
Base Position (Mb)

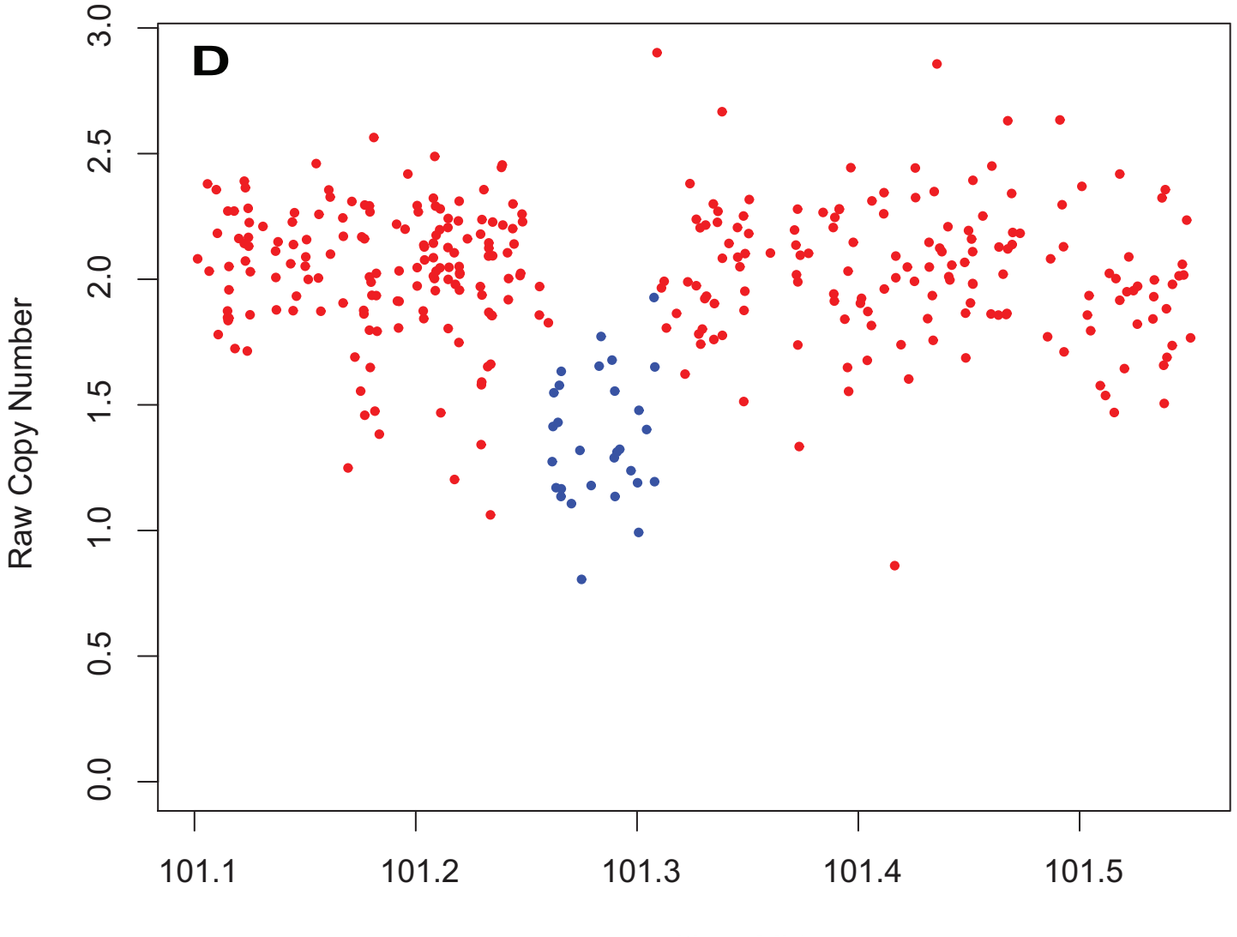

D
Raw Copy Number
Base Position (Mb)

Figure 6

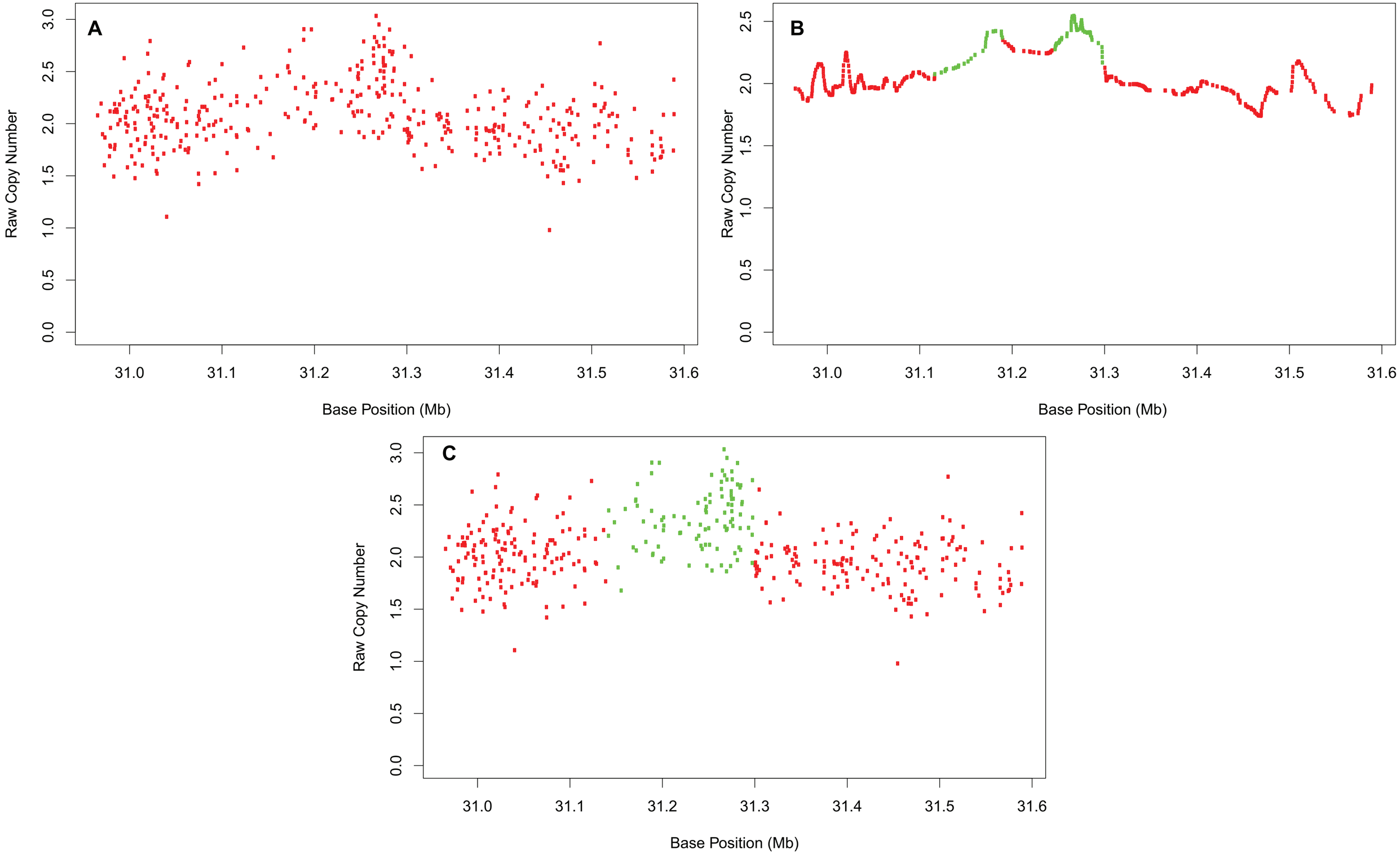

A
B
C
Raw Copy Number
0.0
0.5
1.0
1.5
2.0
2.5
3.0
31.0
31.1
31.2
31.3
31.4
31.5
31.6
Base Position (Mb)

Figure 7

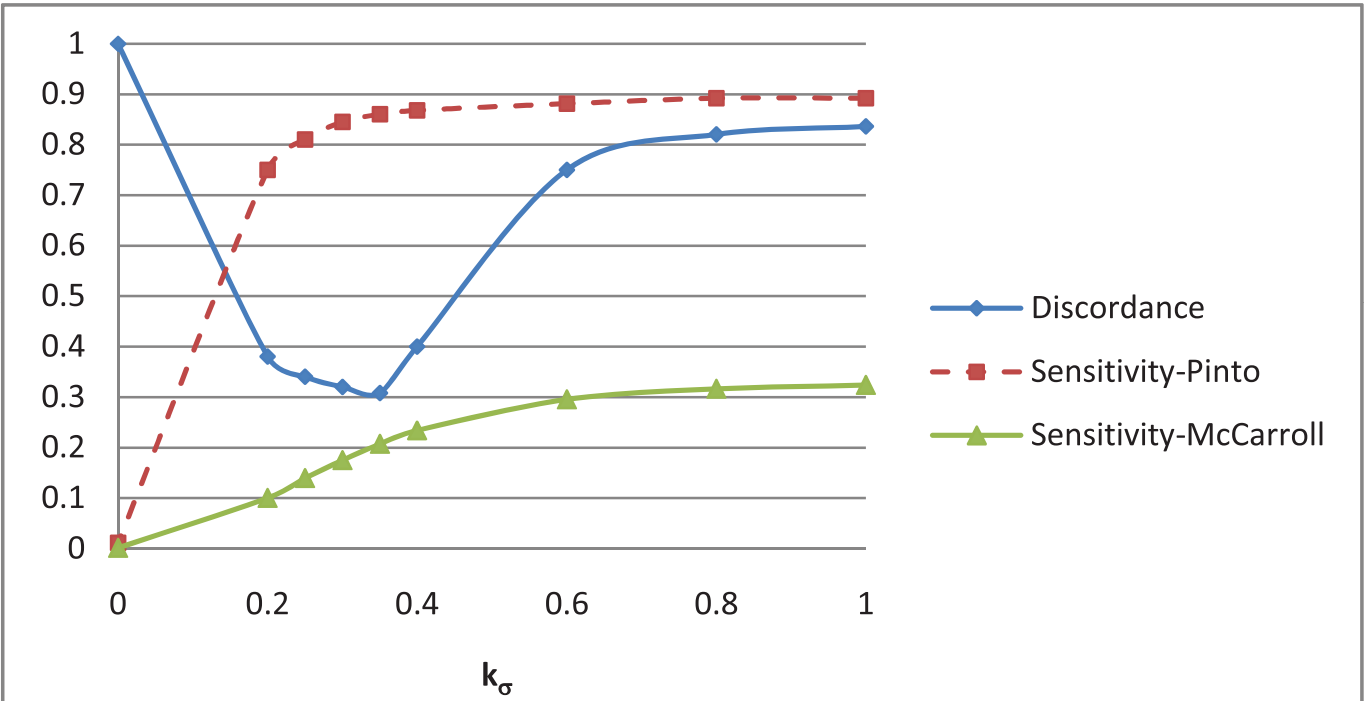
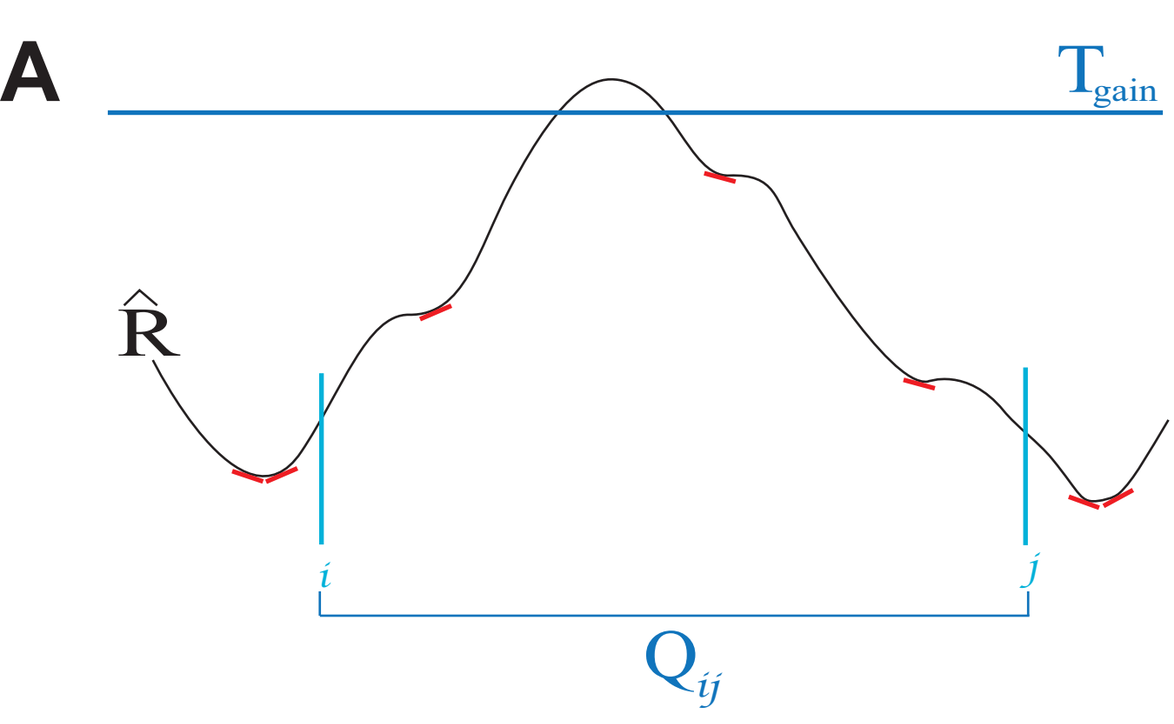
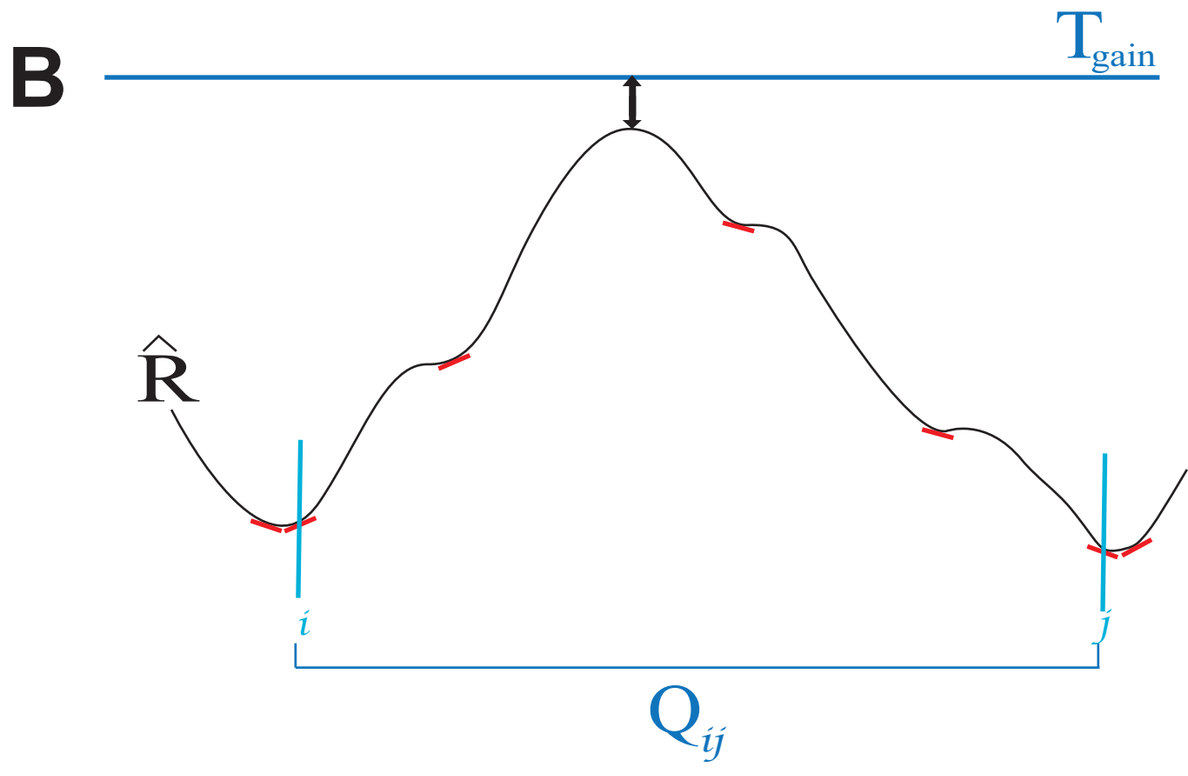
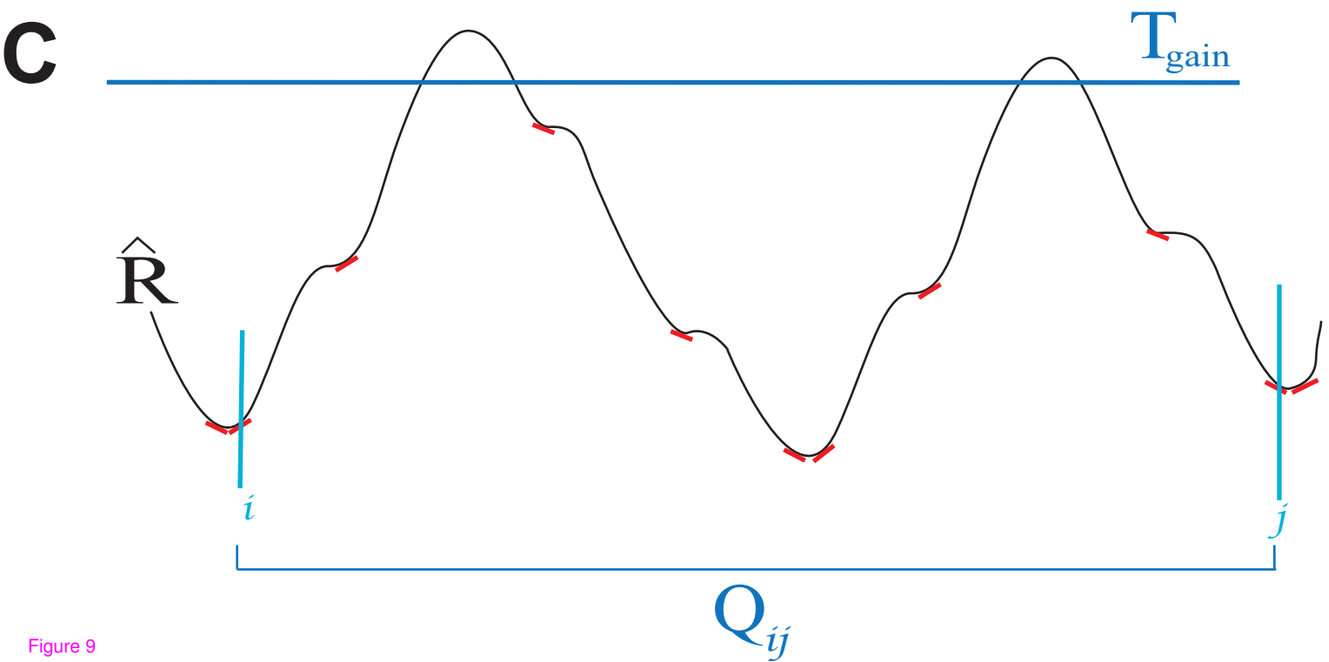

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
k_σ
Discordance
Sensitivity-Pinto
Sensitivity-McCarroll

Figure 8


A
T_{gain}
$\hat{R}$
i
j
Q_{ij}


B
T_{gain}
$\hat{R}$
i
j
Q_{ij}


C
T_{gain}
$\hat{R}$
i
j
Q_{ij}

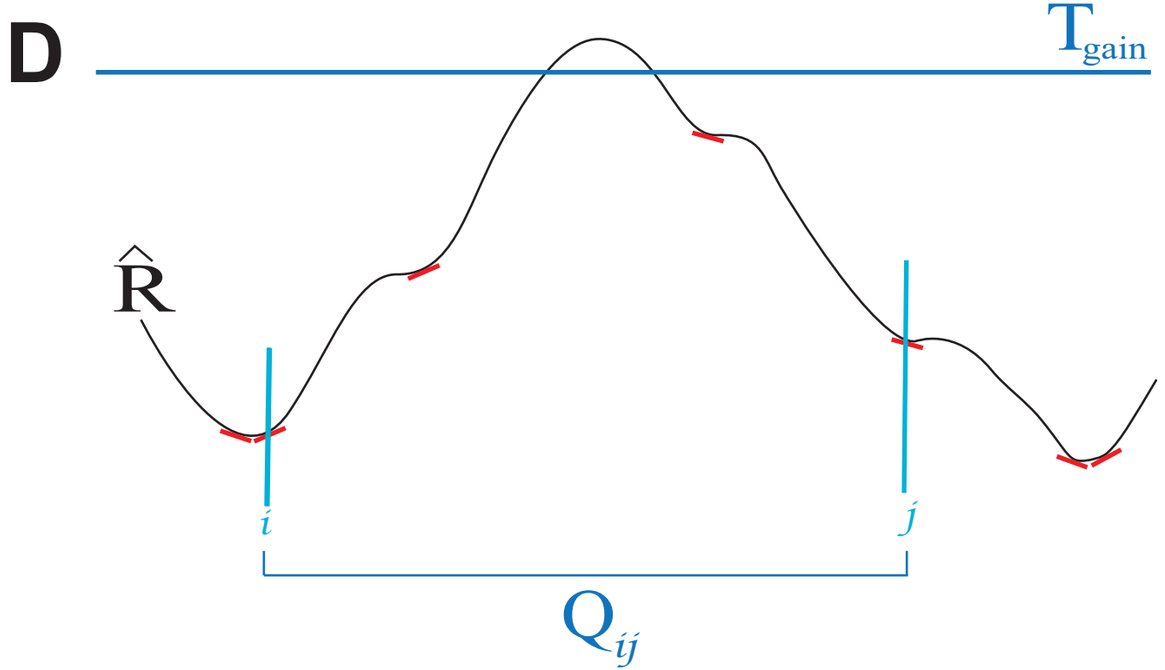

D
T_{gain}
$\hat{R}$
i
j
Q_{ij}

Figure 9

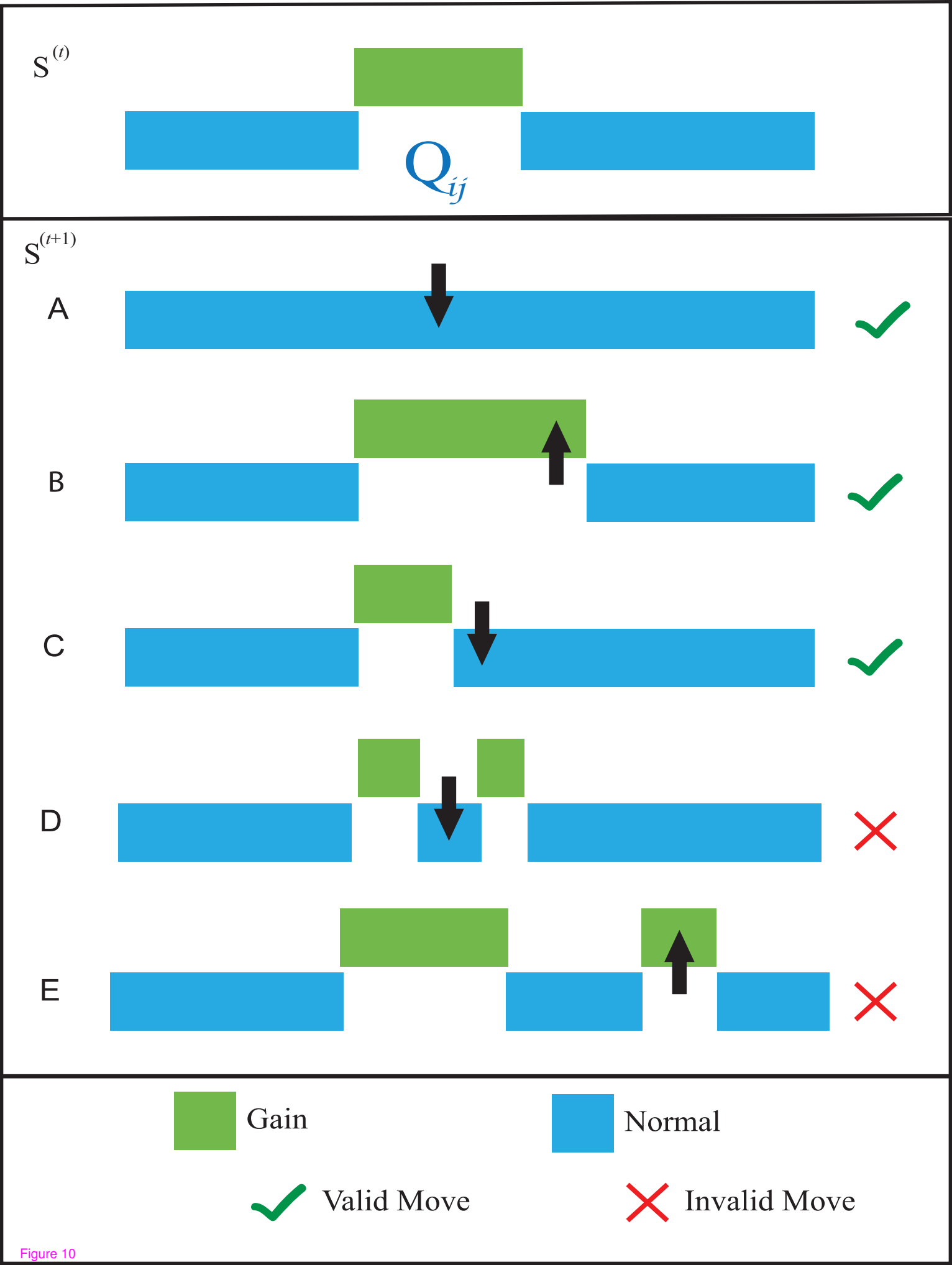


Figure 10