CNVScan: detecting borderline copy number variations in NGS data via scan statistics

E. Bergamini
Institute of Theoretical Informatics
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
elisabetta.bergamini@kit.edu

R. D'Aurizio
IIT-CNR and IFC-CNR
LISM - Laboratory for Integrative Systems Medicine
Via Moruzzi 1, Pisa 56124, Italy
romina.daurizio@gmail.com

M. Leoncini
Dipartimento di Scienze Fisiche, Informatiche e Matematiche
Università di Modena e Reggio Emilia
Via Campi 213/b - 41125 Modena (Italy)
leoncini@unimore.it

M. Pellegrini
IIT-CNR and IFC-CNR
LISM - Laboratory for Integrative Systems Medicine
Via Moruzzi 1, Pisa 56124, Italy
m.pellegrini@iit.cnr.it

ABSTRACT

Background. Next Generation Sequencing (NGS) data has been extensively exploited in the last decade to analyse genome variations and to understand the role of genome variations in complex diseases. Copy number variations (CNVs) are genomic structural variants estimated to account for about 1.2% of the total variation in humans. CNVs in coding or regulatory regions may have an impact on the gene expression, often also at a functional level, and contribute to cause different diseases like cancer, autism and cardiovascular diseases. Computational methods developed for detection of CNVs from NGS data and based on the depth of coverage are limited to the identification of medium/large events and heavily influenced by the level of coverage.

Result. In this paper we propose, *CNVScan* a CNV detection method based on scan statistics that overcomes limitations of previous *read count* (RC) based approaches mainly by being a window-less approach. The scans statistics have been used before mainly in epidemiology and ecology studies, but never before was applied to the CNV detection problem to the best of our knowledge. Since we avoid windowing we do not have to choose an optimal window-size which is a key step in many previous approaches. Extensive simulated experiments with single read data in extreme situations (low coverage, short reads, homo/heterozygoticity) show that this approach is very effective for a range of small CNV (200-500 bp) for which previous state-of-the-art methods are not suitable.

Conclusion. The scan statistics technique is applied and adapted in this paper for the first time to the CNV detection problem. Comparison with state-of-the art methods shows the approach is quite effective in discovering short CNV in rather extreme situations in which previous methods fail or have degraded performance. *CNVScan* thus extends the range of CNV sizes and types that can be detected via read count with single read data.

BCB'15, September 09-12, 2015, Atlanta, GA, USA.

DOI: http://dx.doi.org/10.1145/2808719.2808754.

Categories and Subject Descriptors

J.3 [**LIFE AND MEDICAL SCIENCES**]: Biology and genetics

General Terms

Computational biology

Keywords

Next Generation Sequencing, Copy Number Variation

1. INTRODUCTION

High throughput technologies (HT) are significantly improving the investigation of human variation. The identification of individual genomic variations is usually the first step followed by several downstream analytic tools aiming at inferring causality relationships between the individual genotype and phenotype, including disease conditions. Accurate and correct calling of genomic variations is thus both critical for subsequent analysis, and challenging due to the generally noisy data from HT technologies. Genomic variations may range in size from single nucleotide polymorphisms (SNP) to larger structural variation (SV) as duplications, deletions and translocations of entire chromosomal regions. Here we concentrate on the class of structural variations of size ≥ 50 bp in

length, named *Copy Number Variations*, CNV, (i.e. deletions and duplications). CNV are a particular important form of genomic rearrangements in cancer ([8]), and also a population distribution fingerprint for the general population ([24], [22], [9]). General surveys on several aspects of Structural Variations in genomes and on the challenges faced by detection methods based on Next Generation Sequencing data are in [33] and [4].

We propose a new approach based on scan statistics to detect CNV from Whole Genome Sequencing (WGS) data that we called CNVscan. It works with single-read data and is able to cover an intermediate range of CNV sizes that was not covered in a satisfactory manner by previous approaches. CNVscan produces almost no False Positive and balances well sensitivity and precision (PPV). Also in breakpoint identification, as measured by the Root Mean Square Error (RMSE), produces better results when compared against 4 other state-of-the-art methods. The gap between CNV scan and the alternative methods used in the comparison grows even larger when we consider heterozygous deletions and duplications which are harder to detect. Experiments with synthetic CNV embedded in genomic sequence show that CNVScan is quite superior to other *Read Count* (RC) algorithms for a range of CNV sizes below 1000 (i.e. 200-500), using single-reads, for which the RC approach was deemed not suitable. We term this region of CNV sizes as the "borderline" region as it lies between the region ≥ 1000 usually tackled via RC, and the region of values ≤ 100 which could be tackled with approaches different from read count (e.g. the so-called split reads or pair-end approaches, when pair-end reads are available). Moreover we tested our method and compared performances for the most challenging low-coverage case (5x).

2. STATE OF THE ART

The number of computational methods for CNV detection on NGS data easily exceeds 50, thus for details on individual methods we refer the reader to surveys in [23], [35], [5], [29], [12]. Among the proposed methods for CNV detection, several follow a Read Count (RC) approach (either in a pure or hybrid form) and in [21] [32] one can find an extensive discussion of several technical issues. They are among the most effective for SV in the order of magnitude from about 1000 bp and higher. These methods are almost all based on an initial phase in which the reference genome is split into disjoint buckets of equal length, each aligned read is associated the bucket containing it leftmost bp, and the signal being analyzed is the number of reads in each bucket. Systematic biases due to the sequencing technology, the CG content and the ambiguity in read alignment position (mappability) are usually corrected at this stage.

The bucketing approach has the advantage that the read counts in each bucket can be regarded as *independent* random variables, thus leading to the use of simpler statistical theories to estimate the relevant statistics. On the other hand, the choice of the bucket size is delicate when the size of the bucket, the size of the reads and the size of the CNV are all of the same order of magnitude, since in this case reducing a read to a single point is less justified and border effects of the discretization become noticeable. Each method has different criteria for deciding a preferred window size for a given set of parameters. Interestingly, in [14] it is proposed a general principled approach to the determination of the optimal window size.

Other important parameters are the read length of the library and the average coverage. Having higher coverage benefits almost all methods but it comes at a cost, in particular for whole genome case and when the goal is the screening of a large pool of subjects, thus one should strive to devise methods that work well also with cheaper low-coverage data (about 5x). Over time the attainable read length is slowly increasing as new NGS technologies progress, however methods that are able to exploit reads of shorter length with little quality degradation may have an edge, as producing shorter reads may still be more cost-effective.

In this paper we stay within the read count approach to CNV detection but we dispense with bucketing altogether, relying instead on a classical "scan statistics" approach in order to detect unusual concentrations of reads aligned to the reference genome. Scan statistics tests implicitly and efficiently a continuum range of window sizes and locations and does not use an initial bucketing phase. At the best of our knowledge no CNV detection method has been proposed so far that uses scan statistics as its key model, although such model has been used for a long time in data mining, anomaly detection, epidemiology, ecology, and crime detection. Scan statistics is a statistical technique proposed by J. I. Naus in 1965 [25], and later extended to multidimensional domains by M. Kulldorff in 1997 [16]. Efficient computation of scan statistics is nontrivial thus various efficient approximation algorithms have been proposed [26], [13], [3], [2]. Scan statistics is used in biosurveillance, epidemiology, crime detection and in general for anomalous event detection in space-time data sets [27], [10]. Another area of application is in detection of anomalous activity in fMRI brain imaging data [28].

Background knowledge on CNV distributions is useful also for the realistic simulation of SV in data generators for the purpose of testing detection tools in a more realistic setting [6].

3. CNVSCAN DESCRIPTION

3.1 A novel approach to the detection of CNVs

In this section a novel scan-statistics method for the detection of CNVs is described in detail. Subsection 3.2 gives some background on scan statistics and point Poisson processes and describes the Likelihood Ratio Test used to estimate the parameters and the hypothesis testing, using mainly the notation in [16]. In subsection 3.3 we adapt the abstract setting for generic Poisson processes to the biological problem of detecting CNVs, by aligning reads to the reference and using the depth of coverage. Subsequently, we discuss bias correction steps (subsection 3.4). Finally, having identified the CNV we proceed in subsection 3.5 to call its copy number.

3.2 Scan Statistics and the Likelihood Ratio Test

In general, given a *point process*, many statistical applications deal with the detection of clusters of events. The aim is to determine whether all of the points under study have been generated by a unique baseline distribution, or if there exists a cluster of events that have been generated by a different distribution. Clearly, this classification depends on the probability that the cluster of events is generated by the baseline distribution - i.e. on how unlikely the set of events is, assuming that the baseline distribution generating it. The *Poisson*

process is the simplest type of point process, which can be informally defined as a type of random process for which every realisation consists of a set of isolated points in a given space. More formally, a point process $N = \{N_t : t \geq 0\}$ where N_t (or $N(t)$) denotes the number of arrivals in $(0, t]$ for $t \in [0, +\infty)$ can be defined as a random measure on a complete separable metric space G taking values in the non-negative integers $\mathbb{Z}^+$ (or infinity). In this framework the measure $N(A)$ represents the number of points falling in a subset A of G and if N is a Poisson process it's probability distribution is a Poisson distribution over $[0, t)$ of parameter p, whereas the occurrences are distributed uniformly on any interval.Thus,

$$P[(N(t+\tau) - N(t)) = k] = \frac{e^{-p\tau}(p\tau)^k}{k!} \qquad k = 0, 1, \ldots$$

where $N(t+\tau) - N(t) = k$ is the number of events in time interval $(t, t+\tau]$.
The corresponding process is said an inhomogeneous or homogeneous process if the rate parameter p changes or not over time.
A time-based Poisson process can represent a spatial Poisson process by simply changing the interpretation of the index variable (now x instead of t) in some vector space V (e.g. $\mathbb{R}^2$ or $\mathbb{R}^3$). Here a spatial Poisson process can be defined by the requirement that the random variables that count the number of events inside non-overlapping finite sub-regions A of V, of measure $\mu(A)$, should each have a Poisson distribution and should be independent of each other. So, if we scan the interval $[0, T]$, for $T > 0$ fixed, with a window of size w, $0 < w < T$, we can define:

$$\nu_t = N(t+w) - N(t) \quad \forall t \in [0, T-w]$$

Then,

$$S = S(w, T) = \max_{t \in [0, T-w]} \nu_t$$

is the *scan statistics* for the Poisson process.

In this framework, the null hypothesis is that the underlying distribution for the number of points is Poisson of rate parameter q:

$$N(A) \sim P_0(q\mu(A)) \qquad \forall A \subseteq G$$

where P_0 is the Poisson distribution of parameter $q\mu(A)$:

$$P_0(k) = \frac{e^{-q\mu(A)}(q\mu(A))^k}{k!} \qquad k = 0, 1, \ldots$$

In the alternative hypothesis, conversely, there exists one zone $Z \in \mathscr{Z}$ and a $p > q$ such that

$$N(A) \sim P_0(p\mu(A \cap Z) + q\mu(A \cap Z^C)) \qquad \forall A \subseteq G$$

In words, this means that the points in $A \subseteq Z$ are generated by a Poisson process of rate parameter p, whereas points in $A \subseteq Z^C$ are produced by a Poisson process of rate parameter q.
The likelihood function under the null and under the alternative hypothesis must be computed. Let n_Z denote again the number of points observed in Z and n_G the total number of observed points. We can define the likelihood function under H_0 as:

$$L_0(q) = \frac{e^{-q\mu(G)}(q\mu(G))^{n_G}}{n_G!}$$

which represents the probability of observing n_G points inside G, supposing that they were all generated by a Poisson process of parameter q.
Again, the maximum likelihood estimator $\hat{q}_0$ for q under the null hypothesis can be computed as the value of q that maximizes $L_0(q)$. Doing some algebra,

$$\hat{q}_0 = \frac{n_G}{\mu(G)}$$

Analogously, we can define the likelihood function under the alternative hypothesis as:

$$L_1(Z, p, q) = A(Z, p, q)B(Z, p, q)$$

$$A(Z, p, q) = \frac{e^{-p\mu(Z)}(p\mu(Z))^{n_Z}}{n_Z!}$$

$$B(Z, p, q) = \frac{e^{-q(\mu(G)-\mu(Z))}(q(\mu(G)-\mu(Z)))^{n_G-n_Z}}{(n_G-n_Z)!}$$

which represents the joint probability of observing n_Z points in Z and $n_G - n_Z$ points outside Z, under the assumption that the points in Z follow a Poisson distribution of parameter p and the points outside Z follow a Poisson distribution of parameter q.
The maximum likelihood estimators $\hat{p}_{1,Z}$, $\hat{q}_{1,Z}$ under H_1 for a fixed Z are:

$$\hat{p}_{1,Z} = \frac{n_Z}{\mu(Z)} \qquad \hat{q}_{1,Z} = \frac{n_G - n_Z}{\mu(G) - \mu(Z)}$$

if $\frac{n_Z}{\mu(Z)} > \frac{n_G - n_Z}{\mu(G)-\mu(Z)}$, and

$$\hat{p}_{1,Z} = \hat{q}_{1,Z} = \frac{n_G}{\mu(G)}$$

otherwise.
Then, calling $L_1(Z) = L_1(Z, \hat{p}_{1,Z}, \hat{q}_{1,Z})$, the estimator Z' is defined as follows:

$$\hat{Z} = \{Z : L_1(Z) \geq L_1(Z') \quad \forall Z' \in \mathscr{Z}\}$$

and it represents the most-likely cluster of the observed dataset.
We define the likelihood ratio λ as:

$$\lambda = \frac{\sup_{Z \in \mathscr{Z}, p > q} L_1(Z, p, q)}{L_0} = \frac{L_1(\hat{Z})}{L_0}$$

where L_0 is defined as

$$L_0 \stackrel{\text{def}}{=} L_0(\hat{q}_0)$$

.
In this case, doing some algebra [16], λ can be rewritten as

$$\lambda = \sup_{Z \in \mathscr{Z}} \frac{\left(\frac{n_Z}{\mu(Z)}\right)^{n_Z} \left(\frac{n_G - n_Z}{\mu(G)-\mu(Z)}\right)^{n_G - n_Z}}{\left(\frac{n_G}{\mu(G)}\right)^{n_G}} I_Z \qquad (1)$$

with

$$I_Z = I\left(\frac{n_Z}{\mu(Z)} > \frac{n_G - n_Z}{\mu(G) - \mu(Z)}\right), \qquad (2)$$

where $I(.)$ is the characteristic function of a predicate, and we assume that there is at least one Z such that $\frac{n_Z}{\mu(Z)} > \frac{n_G - n_Z}{\mu(G)-\mu(Z)}$. Otherwise $\lambda = 1$.
Notice that so far only cluster with an abnormally high number of cases have been considered, but the same procedure

can be used also to detect clusters with a particularly low number of points in a Poisson process. This can be accomplished by simply computing the maximum likelihood estimators $\hat{p}$, $\hat{q}$ conditioned on Z over $p < q$, instead of $p > q$, obtaining again equation (1) but with

$$I_Z = I\left(\frac{n_Z}{\mu(Z)} < \frac{n_G - n_Z}{\mu(G) - \mu(Z)}\right). \quad (3)$$

Apart from the simplest cases, exact derivations of the distribution of λ are impracticable [16]. For this reason, Monte Carlo simulations are usually employed to give good approximations of the test statistics [17]. The simulations consist in generating several datasets under the null hypothesis and in computing for each of them the value of the test statistics λ. In particular for us, the datasets are Poisson processes of rate parameter equal to the maximum likelihood estimator $\hat{q}_0$ under the null hypothesis. The p-value is then approximated with the fraction of datasets whose test statistics is higher than the one observed in the original dataset.

The inference process consists, first, in setting a significance level α, then the maximum likelihood estimator under the null hypothesis $\hat{q}_0$ is computed for the given dataset, as well as L_0 and $L_1(Z)$ for all the possible subsets Z in the dataset, in order to find the test statistics λ. Then N datasets are generated under H_0, λ_i is computed for each of them and the p-value is approximated as the fraction of datasets i whose λ_i is greater than λ. Then, H_0 or H_1 is accepted according to α.

3.3 Likelihood ratio test for genomic data

All read-depth methods rely on the assumption that the reads are sampled uniformly over the sequenced genome. Under this assumption, the number of reads whose initial position is a certain base of the reference genome follows a Poisson distribution [21]. Besides the number of reads that start in a certain position of the reference, another quantity that can be considered is the *coverage*. The coverage of a base of the reference genome is the number of aligned reads *overlapping* that specific base.

While the number of reads that start in a certain position of the reference can be assumed to follow a Poisson distribution, the coverage rigorously could not. Indeed, the coverage of a base can not be considered to be independent from those of the neighbouring bases, since if one read overlaps a position of the reference, it "probably" overlaps also the adjacent positions. However, the length of the reads is extremely short compared to the whole reference genome, so the coverage of each base is actually independent from those of the greatest majority of the other bases. For this reason, the novel read-depth approach we developed makes use of the likelihood ratio test applied to the coverages of the bases.

As previously described, the likelihood ratio test for a Poisson process contemplates the computation of a test statistics λ, whose formulation is in equations (1) and (2) for amplifications and equations (1) and (3) for deletions.

Within the framework of CNVs detection, the whole space G can be seen as the reference genome. Therefore, its measure $\mu(G)$ is the number of bases composing the reference and the number of points n_G is the sum of the coverages of all the bases of the reference.

In this context, the possible subsets Z of G are consecutive sub-strings of the reference and, analogously, the measure of a sub-string Z is the number of bases composing Z and n_Z is the sum of the coverages of such bases.

The above-described algorithm aims to detect areas with a particularly high coverage, therefore it is only suitable for duplications detection. However, a small modification can be done in order to scan for sub-strings with a particularly low coverage. Recalling the considerations made in Section 3.2, when scanning for events with a lower number of points, the function $\lambda(Z)$ to maximize over the possible Z is

$$\lambda(Z) = \frac{\left(\frac{n_Z}{\mu(Z)}\right)^{n_Z}\left(\frac{n_G - n_Z}{\mu(G)-\mu(Z)}\right)^{n_G - n_Z}}{\left(\frac{n_G}{\mu(G)}\right)^{n_G}}$$

if $\frac{n_Z}{\mu(Z)} < \frac{n_G - n_Z}{\mu(G)-\mu(Z)}$, and $\lambda(Z) = 1$ otherwise.

3.4 Preprocessing: bias correction

Although the read sampling process is supposed to be approximatively uniform over the test genome, two main sources of bias affect the real distribution of the coverage: the local GC content and the genomic mappability. The first one represents the percentage of guanines and cytosines of a certain area of the reference. It has been shown that the GC content heavily influences the read counts: for both particularly high or particularly low values of GC content, the coverage is significantly lower than the average [31]. Conversely, mappability bias is due to the fact that the genome contains many repetitive elements and aligning reads to these positions leads to ambiguous mapping.

In order to improve the precision of the algorithm, a bias-correction phase is performed on the distribution of the coverages before the actual detection process. The following sections describe the bias-correction techniques that were adopted in our work.

3.4.1 GC content

In analogy with a commonly-used normalization technique for read-depth methods [37], we adjusted the coverages by using the observed deviation in coverage for every possible GC percentage. More specifically, for each possible position k of the reference genome, we considered a window w_k of a fixed length centred in k and we computed the GC percentage (number of guanines and cytosines in the window w_k divided by total number of bases). Then, for each possible GC percentage $(0, 1, 2, \ldots, 100\%)$, we computed the average coverage of the positions with that specific GC percentage and we corrected the coverages according to the following formula:

$$\tilde{c}_k = c_k \frac{\bar{m}}{m_{GC(k)}} \qquad k = 1, 2, \ldots, n_G$$

where c_k is the coverage of the k^{th} base of the reference, $\bar{m}$ is the average coverage along the reference, $m_{GC(k)}$ is the average coverage within the positions with the same GC content as k and $\tilde{c}_k$ is the normalized coverage.

The window should represent the area of the reference that actually influences the coverage with its GC content. Several studies have been devoted to the understanding of the GC bias, however the causes that lead to such bias and the exact distribution of the coverage as a function of the GC content are still unknown. However, recent studies [7] have shown that it is the GC content of the full DNA fragment, not only the sequenced read, that most influences the coverage. For

this reason, in our work we set the length of the window to a value that is approximately equal to the sampled fragments.

3.4.2 *Mappability*

To correct the mappability bias, existing approaches usually give a mappability score to genomic sequences, representing how unique the sequence is within the reference genome. While the majority of existing methods divide the reference into non-overlapping windows, our approach requires the computation of a mappability score for each possible position of the genome. As we did for the GC content, this can be accomplished by considering windows of fixed length centred in each possible position of the reference. Therefore, for each position k, we want to find out how unique the sequence w_k centred in k is within the reference or, in other words, the number of occurrences of w_k along the reference. Theoretically, various methods can be employed to compute the number of occurrences of all the w_k. The simplest one is a brute-force approach, consisting in the explicit enumeration and counting of all the w_k present in the reference; however, given the dimension of the genomic data, this approach is usually unfeasible if we want to allow mismatches between w_k and the occurrences. Also using existing mapping algorithms to map each w_k becomes impracticable if mismatches are taken into account. In particular, the speed of all implementations of mapping algorithms (that allow mismatches) always shows some dependency on the number of matches found in the reference [11]; this means that aligning a certain number of reads that all map to thousands of locations in the genome will be much slower than aligning the same number of reads that map uniquely. Therefore, most of the degradation in performance actually comes from the fraction of w_k showing high frequencies (i.e., number of occurrences), where sometimes thousands of sequences which are equivalent to w_k (within the specified number of mismatches) exist in the reference.
In [11], this problem is taken into account by performing an approximation on the exact number of matches of sequences with high frequencies. The approximation is the following: each time a certain w_k is mapped within the given number of mismatches to a set $\mathcal{S}$ of sequences of the reference, one can pretend that all the sequences in $\mathcal{S}$ have already been mapped, assign to them a frequency value equal to the number of elements in $\mathcal{S}$, and skip all of them from that point on. Such a strategy is not enough to completely remove the computational-cost problem - it is effective in eliminating only the equivalent sequences occurring in the reference after w_k has been mapped -, and is only exact when no substitutions are allowed. From a practical standpoint, however, the speed results considerably improved and the mappabilities computed in this way are good approximations of the exact values.

3.5 Postprocessing: copy-number estimation

Once the candidate variation $\hat{Z}$ that maximizes $\lambda(Z)$ has been found, it has to be decided whether to accept the null hypothesis H_0 (no variation has occurred) or to reject it (a variation has occurred). As described in Section 3.2, this requires the computation of a p-value using Monte Carlo simulations. In these simulations, a number N sufficiently large of datasets are created under H_0. More specifically, instead of simply generating N Poisson processes of rate parameter $\frac{n_G}{m_G}$, in each dataset a real sampling process is simulated, thus taking into account also the reads' length.

If the p-value is smaller than a significance level α, the null hypothesis is rejected, otherwise it is accepted. When H_0 is rejected, it is then necessary to estimate the copy number of the detected variation. In case of data from haploid regions, this can be obtained by approximating to the nearest integer the ratio

$$r = \frac{\bar{c}_{\hat{Z}}}{\bar{c}_{G-\hat{Z}}}$$

between the average coverage $\bar{c}_{\hat{Z}}$ inside the region $\hat{Z}$ and the average coverage $\bar{c}_{G-\hat{Z}}$ outside $\hat{Z}$. In case of data from diploid regions, the copy number can be obtained in an analogous way normalizing to copy number 2, i.e. multiplying r by 2. This simple estimation strategy follows the assumption that the sequencing process is uniform and consequently the number of reads that map to a genomic region is expected to be proportional to the number of times the region appears in the DNA sample.

3.6 Two-level heuristic for gain in speed

Given the large size of sequencing datasets, speeding up the variant-detection process is an issue of great importance. Finding the sub-string that maximizes the discrepancy function requires the computation of $\lambda(Z)$ on all possible subsequences Z of the reference, so in general there is no faster way to find the optimal solution than the one previously described. However, it is possible to skim the space of the substrings, by identifying a zone where the number of reads is particularly high (or low, in the case of deletions) and then computing the discrepancy function only on the sub-sequences of that zone. We apply a heuristics that conglomerates the positions of the reference in windows of fixed length w and computes the discrepancy function on all the groups of adjacent windows, in order to find the zone with maximum discrepancy function. Once this zone ($\hat{Z}_0$ has been found, the precision of the candidate-variation coordinates is improved by re-computing the discrepancy function in all the sub-strings around $\hat{Z}_0$. More precisely, the heuristic considers all the sub-string of the reference whose initial position is within the first window composing $\hat{Z}_0$, or the immediately preceding one, and whose ending position is within the last window composing $\hat{Z}_0$, or the following one (see Figure 1). Naturally, instead of considering all possible groups of adjacent windows, it is possible to compute the discrepancy function only on groups whose length is within a certain interval. In this manner, it is possible to reduce the computational time required by the algorithm, in case we know a range for the possible length of the SV to be detected.

4. EXPERIMENTS

4.1 Data Generation

The simulation pipeline uses several standard data manipulation tools. We performed the experiments described in this section on a stretch of 1M bases in the human chromosome 21 of the hg19 reference genome. Given the limited effects of mappability issues in this relatively short region, we did not apply the correction outlined in Section 3.4.2. Starting from such region, we:

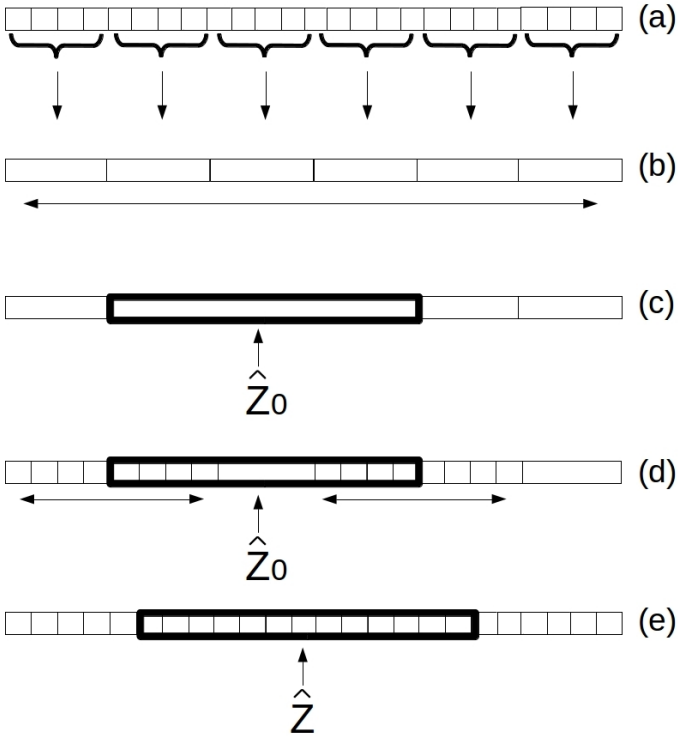

Figure 1: Heuristic for the computation of the sub-string with maximum discrepancy function. First (a), the positions of the reference genome are conglomerated in windows of fixed length w, creating a new vector containing for each window the sum of the coverages of the positions inside that window. Then (b), the discrepancy function is computed on every sub-sequence of the new vector, leading to candidate variation $\hat{Z}_0$ (c). Then, to improve the precision of the detection, the discrepancy function is recomputed on all the sub-string of the reference whose initial position is within the first window of $\hat{Z}_0$, or the immediately preceding one, and whose ending position is within the last window of $\hat{Z}_0$, or the following one (d). The sub-string that maximizes the discrepancy function is then the final candidate variation $\hat{Z}$ (e).

a) inject in random positions a set of CNV of various sizes and types (deletions/duplications) using RVSSim [6].

b) simulate the Illumina sequencing using the ART tool [15] over the sequence obtained at a) extended of 200K bp on both sides.

c) perform alignments of these reads using BWA [18] over the reference h19 (limited to the 1.4M bp region defined in b).

d) use Samtools [19] to generate an ordered and indexed BAM file

The output of this pipeline is the raw data to be given as input to the CNV calling software. Some of these tools work in a comparative mode and need a reference set of reads. The reference is obtained from the same pipeline described above, just omitting step a).

4.2 CNV Generation

The generation of the SV uses a blueprint from [34]. We selected three random values in the range [200-500], we associate to each such value randomly the label "Dup" for duplication or "Del" for a deletion. Moreover randomly we select a location in the chosen 1M bp stretch. The RVSSim tool ensures that these locations are not overlapping. This data generation phase is repeated 32 times for each parameters setting and the measures in the subsection 4.4 are the averages of these 32 runs.

4.3 Competitors and parameter optimization

We compare the performance of CNVscan against that of these 4 methods: BIC-Seq [34], CNVnator [1], JointSLM [20], and CNV-seq [36], which are among the most cited in the literature adopting the RC approach for WGS. We have used mostly the default parameters. When the default parameters did not return any result we relaxed them in order to be more permissive. In these cases, for CNVNator we adjusted the bin size according to the recipes reported in the supplementary material of [1] while for CNV-seq we set the p-value threshold to 0.05. JointSLM can handle a pool of samples, but, for these experiments it has been run with the option for a single sample.

4.4 Quality Measures

Given a set of predicted CNV and a set of embedded CNV represented as intervals we can group (and count) the nucleotides into four classes according to the prediction label, and to the fact that it corresponds to a real CNV or not.

a) nTP is the number of nucleotide labeled $Positive$, that are really part of a CNV.

b) nFP is the number of nucleotide labeled $Positive$, that are not really part of a CNV.

c) nTN is the number of nucleotide labeled $Negative$, that are not really part of a CNV.

d) nFN is the number of nucleotide labeled $Negative$, that are really part of a CNV.

Standard quality measures are the Sensitivity (Sn):

$$Sn = \frac{nTP}{nTP + nFN} \tag{4}$$

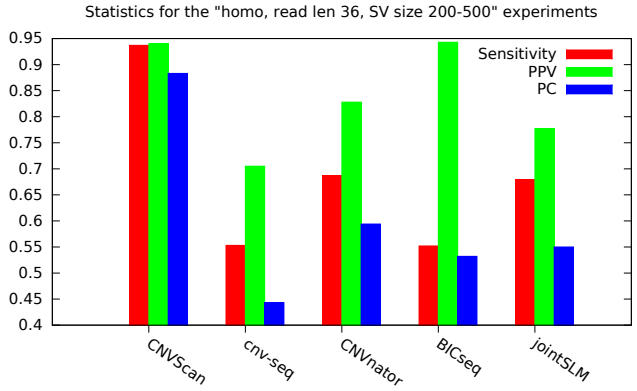

Figure 2: PC, PPV and Sensitivity for 5 algorithms.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	**32**	**0**	**1**	**0.937**	0.940	**0.883**	**26**
cnv-seq	20	2	65	0.553	0.705	0.443	81
CNVnator	31	**0**	32	0.687	0.828	0.594	70
BICseq	25	**0**	49	0.552	**0.943**	0.532	32
jointSLM	**32**	5	28	0.679	0.777	0.550	81

Table 1: Homozygote CNV. Read length 36, Mean Coverage 5, SV size range 200–500 bp.

and the Positive Predicted Value (PPV):

$$PPV = \frac{nTP}{nTP + nFP} \tag{5}$$

A balanced measure has been proposed by Pevzner and Sze [30], called the *Performance Coefficient* (abbr. *PC*) and defined as:

$$PC = \frac{nTP}{nTP + nFN + nFP}. \tag{6}$$

As a single index the Performance Coefficient (PC) is the most informative, as it balances well PPV and Sensitivity.

Results of experiments for borderline SV (size range 200-500) are shown in Tables 1, 2, 3 and 4. Figures 2, 3, 4, 5 report the same data for PC, PPV and Sensitivity in graphical form.

In each table we report: 1) the number nF of data sets (out of 32) for which the algorithm provides at least one prediction; 2) the number of FP of false positive and FN of false negatives predictions out of 32 x 3 = 96 embedded SV, where we count a hit if the intersection of a prediction with an embedded SV is not empty; 3) the sensitivity, Positive Predicted Value and Performance Coefficient at the base level as defined above; 4) the root mean square error ($RMSE$) of the distance of the endpoints of each embedded SV to the endpoints of the best prediction. For nF, FP, FN, and $RMSE$ lower values indicate better quality. For PC, PPV and Sensitivity, higher values indicate better quality.

Results of experiments for SV in the size range 1000-20000 are shown in Tables 5, 6, 7 and 8.

4.5 Time

We conducted experiments on a Server with 4 cpu Intel Xeon (8 cores) and 256 Gb RAM, with operating system Red Hat Enterprise Linux Server v. 5 (64 bits). The CNVScan prototype code is written in Python 2.7, and, at the moment, it

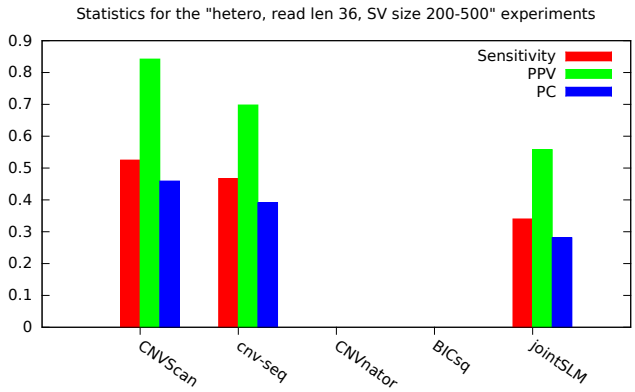

Figure 3: PC, PPV and Sensitivity for 5 algorithms. Missing columns indicate no output.

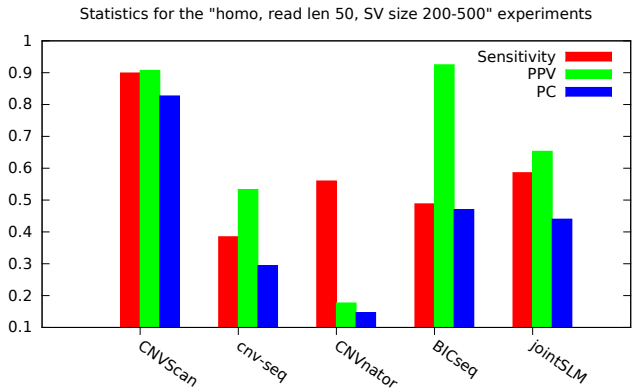

Figure 4: PC, PPV and Sensitivity for 5 algorithms .

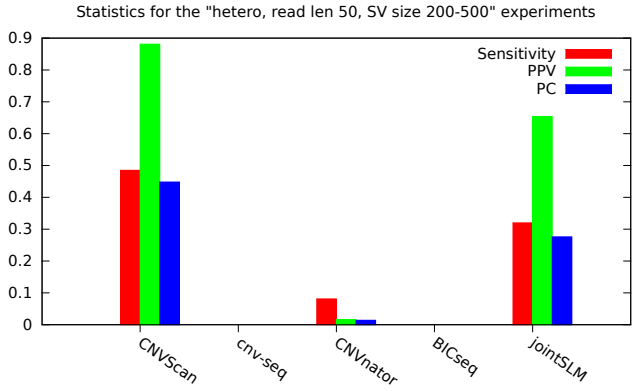

Figure 5: PC, PPV and Sensitivity for 5 algorithms. Missing columns indicate no output.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	**25**	**0**	**58**	**0.525**	**0.842**	**0.459**	**83**
cnv-seq	6	**0**	88	0.469	0.698	0.391	108
CNVnator	0	0	96	-	-	-	-
BICseq	0	0	96	-	-	-	-
jointSLM	18	8	77	0.340	0.558	0.281	**83**

Table 2: Heterozygote CNV. Read length 36, Mean Coverage 5, SV size range 200–500 bp.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	**32**	1	**6**	**0.899**	0.907	**0.827**	**32**
cnv-seq	10	1	86	0.385	0.533	0.293	150
CNVnator	30	49	48	0.560	0.177	0.147	535
BICseq	25	**0**	56	0.488	**0.925**	0.470	39
jointSLM	27	8	45	0.586	0.653	0.440	83

Table 3: Homozygote CNV. Read length 50, Mean Coverage 5, SV size range 200–500 bp.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	19	**0**	**69**	**0.485**	**0.881**	**0.448**	**49**
CNVseq	0	0	96	-	-	-	-
CNVnator	**25**	53	90	0.081	0.016	0.014	1271
BICseq	0	0	96	-	-	-	-
jointSLM	18	4	78	0.320	0.654	0.276	74

Table 4: Heterozygote CNV. Read length 50, Mean Coverage 5, SV size range 200–500 bp.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	32	0	2	**0.996**	**0.997**	**0.993**	**35**
cnv-seq	32	0	3	0.985	0.984	0.969	164
CNVnator	32	0	2	0.995	0.991	0.986	65
BICseq	32	0	0	0.991	0.994	0.985	2915
jointSLM	32	1	0	0.993	0.995	0.988	66

Table 5: Homozygote CNV. Read length 36, Mean Coverage 5, SV size range 1000–20000 bp.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	32	0	4	**0.990**	0.994	**0.984**	**72**
cnv-seq	30	0	21	0.538	**0.997**	0.537	4499
CNVnator	32	0	2	0.985	0.992	0.978	152
BICseq	32	0	4	0.978	0.981	0.960	264
jointSLM	32	10	2	0.983	0.977	0.961	153

Table 6: Heterozygote CNV. Read length 36, Mean Coverage 5, SV size range 1000–20000 bp.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	32	1	3	**0.997**	0.976	0.970	**40**
cnv-seq	32	0	6	0.973	0.974	0.948	258
CNVnator	32	0	4	0.990	**0.996**	**0.986**	51
BICseq	32	0	2	0.989	0.995	0.984	1182
jointSLM	32	2	1	0.989	0.993	0.982	100

Table 7: Homozygote CNV. Read length 50, Mean Coverage 5, SV size range 1000–20000 bp.

is not optimized for parallel execution. In this setting each run of CNVScan takes about 3 minutes. As CNVScan has ample scope for exploiting data level parallelism, we reckon that substantial speed ups can be further attained on multi-core/multi-processor architectures, thus code optimization is left as future work.

Method	nF	FP	FN	Sn	PPV	PC	RMSE
CNVScan	32	0	3	**0.990**	0.993	**0.983**	**103**
cnv-seq	30	0	28	0.545	**0.996**	0.542	4216
CNVnator	32	0	6	0.969	0.987	0.959	188
BICseq	32	0	8	0.958	0.975	0.935	334
jointSLM	32	9	2	0.981	0.973	0.955	227

Table 8: Heterozygote CNV. Read length 50, Mean Coverage 5, SV size range 1000–20000 bp.

5. DISCUSSION

On a global outlook our experiments are in line with results reported in literature for read count methods (see e.g. those in [34]. All 5 tested methods perform better in the standard CNV size range (above 1000), than in the borderline size range (200-500). In particular summing up the values in tables 5, 6, 7, 8, and comparing them with those in tables 1, 2, 3, 4, we obtain that out of 640 runs for a total of 636 runs the 5 algorithms returned at least one prediction (vs 375/640), obtaining in total 23 false positives (vs. 131), and 103 false negatives out of 1920 CNV embedded (vs. 1260/1920). This confirms that that borderline case is indeed tougher on average for the RC algorithms. Thus we will discuss the two size ranges separately.

Normal range : above 1000. See tables 5, 6, 7, 8. Here the three standard normalized measures (Sn, PPV and PC) are all very high above 0.9 in all cases (except for cnv-seq on heterozygote CNV where we notice a drop in Sn and PC due to a high number of false negatives. In terms of comparative ranking the proposed CNVscan has the best performance for 8 measures (out of 12), cnv-seq for 2 , and CNVnator for 2.
For this class of CNV sizes a second discriminating measure is the precision in detecting the CNV breakpoints, which we measure as the RMSE of the end point distances for a prediction with maximal non null overlap against the embedded CNV. For this measure CNVscan ranks first in 4 cases out of 4. Looking at the ability to edge false positives vs false negatives, we notice that CNVscan has a trade-off (1 fp, 12 fn) that is quite close to that of CNVnator (0fp, 14 fn), and BICseq (0fp, 14fn). CNV-seq also attains 0 fp, but with a large increase in false negatives (58 fn). At the opposite side of the spectrum, JointSLM has low number of false negatives (5 fn) but at the cost of introducing a larger number of false positives (22 fp).
For this size range class CNVscan thus performs marginally better than, or equally to, the other 4 algorithms used in the comparison in most of the chosen quality measures, in a consistent way.

Borderline range : from 200 to 500. For this size class the story told by Figures 2, 3, 4, 5 and Tables 1, 2, 3, 4 is more dramatic. First we notice a sharper distinction between the cases of homozygote and heterozygote CNV. Depending also on the read size, for the heterozygote case (see Tables 2, 4, and Figures 3, 5) CNVSseq, CNVnator and BICseq may fail to return any prediction at all, thus making them virtually not comparable. Only jointSLM has consistent performances that can be compared. CNVscan has fewer false positives (0

vs. 12) and fewer false negatives (127 vs 155) thus showing a better balanced behavior. In the normalized measures (Sn, PPV, and PC) CNVScan leads over jointSLM in 6 measures (out of 6), often by a margin higher than 50%.
For the homozygote case (see Tables 1, 3, and Figures 2, 4), with some variability, all methods make predictions in a sufficient number of experiments. CNVscan attains a better fp/fn trade-off (1 fp, 7 fn), with respect to cnv-seq (3 fp, 151 fn), CNVnator (49fp, 80fn) and BICseq (0 fp, 105 fn), and jointSLM (13 fp, 73 fn). This better balancing is reflected in the ranking of the normalized measures (Sn, PPV, PC) where CNVscan ranks first in 4 measures out of 6. BICseq ranks first in the remaining 2 measures, namely PPV, because of its very conservative calling policy (however CNVscan is a close second). The RMSE measure is also consistently better for CNVScan. This set of experiments shows that in the borderline CNV size range CNVscan is superior to the 4 state of the art read count based methods by a remarkably high margin in most measures.
Conclusions. For the homozygote CNVs, CNVscan is the only one among the methods shown here that comes close to bridging the gap between the borderline and the standard size range for RC methods, as measured, say, by the PC measure. For the heterozygote CNVs, the performance of CNVscan (for these low coverage, small size, weak signal tests) degrades tough it is retaining a better detection capability w.r.t competing methods.
The current implementation of CNVscan is optimized for single read data. We plan to extend it to handling also pair-end data, thus making it possible to compare its performance also with other methods that exploit pair-end data (either based on split reads signals, or on pair-end distance signals). We also plan to test our method on non-synthetic validated benchmark sequencing data.

6. ACKNOWLEDGMENTS

The present work is partially supported by the Flagship project InterOmics(PB. P05), funded by the Italian Ministry for Instruction University and Research (MIUR) and CNR organizations, and by the joint IIT-IFC Laboratory of Integrative Systems Medicine (LISM).

7. REFERENCES

[1] A. Abyzov, A. E. Urban, M. Snyder, and M. Gerstein. Cnvnator: an approach to discover, genotype, and characterize typical and atypical cnvs from family and population genome sequencing. *Genome research*, 21(6):974–984, 2011.

[2] D. Agarwal, A. McGregor, J. M. Phillips, S. Venkatasubramanian, and Z. Zhu. Spatial scan statistics: approximations and performance study. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 24–33. ACM, 2006.

[3] D. Agarwal, J. M. Phillips, and S. Venkatasubramanian. The hunting of the bump: on maximizing statistical discrepancy. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1137–1146. Society for Industrial and Applied Mathematics, 2006.

[4] C. Alkan, B. P. Coe, and E. E. Eichler. Genome structural variation discovery and genotyping. *Nature Reviews Genetics*, 12(5):363–376, 2011.

[5] A. Alkodsi, R. Louhimo, and S. Hautaniemi. Comparative analysis of methods for identifying somatic copy number alterations from deep sequencing data. *Briefings in bioinformatics*, page bbu004, 2014.

[6] C. Bartenhagen and M. Dugas. Rsvsim: an r/bioconductor package for the simulation of structural variations. *Bioinformatics*, page btt198, 2013.

[7] Y. Benjamini and T. P. Speed. Summarizing and correcting the gc content bias in high-throughput sequencing. *Nucleic acids research*, page gks001, 2012.

[8] P. J. Campbell, P. J. Stephens, E. D. Pleasance, S. O'Meara, H. Li, T. Santarius, L. A. Stebbings, C. Leroy, S. Edkins, C. Hardy, et al. Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nature genetics*, 40(6):722–729, 2008.

[9] D. F. Conrad, D. Pinto, R. Redon, L. Feuk, O. Gokcumen, Y. Zhang, J. Aerts, T. D. Andrews, C. Barnes, P. Campbell, et al. Origins and functional impact of copy number variation in the human genome. *Nature*, 464(7289):704–712, 2010.

[10] K. Das, J. Schneider, and D. B. Neill. Anomaly pattern detection for biosurveillance. *Advances in Disease Surveillance*, 5:19, 2008.

[11] T. Derrien, J. Estellé, S. M. Sola, D. G. Knowles, E. Raineri, R. Guigó, and P. Ribeca. Fast computation and applications of genome mappability. *PloS one*, 7(1):e30377, 2012.

[12] J. Duan, J.-G. Zhang, H.-W. Deng, and Y.-P. Wang. Comparative studies of copy number variation detection methods for next-generation sequencing technologies. *PloS one*, 8(3):e59128, 2013.

[13] J. Glaz, J. I. Naus, S. Wallenstein, S. Wallenstein, and J. I. Naus. *Scan statistics*. Springer, 2001.

[14] A. Gusnanto, C. C. Taylor, I. Nafisah, H. M. Wood, P. Rabbitts, and S. Berri. Estimating optimal window size for analysis of low-coverage next-generation sequence data. *Bioinformatics*, 30(13):1823–1829, 2014.

[15] W. Huang, L. Li, J. R. Myers, and G. T. Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[16] M. Kulldorff. A spatial scan statistic. *Communications in Statistics-Theory and Methods*, 26(6):1481–1496, 1997.

[17] M. Kulldorff. Spatial scan statistics: models, calculations, and applications. In *Scan Statistics and Applications*, pages 303–322. Birkhauser, Boston, 1999.

[18] H. Li and R. Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[19] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[20] A. Magi, M. Benelli, S. Yoon, F. Roviello, and F. Torricelli. Detecting common copy number variants in high-throughput sequencing data by using JointSLM algorithm. *Nucleic acids research*, page gkr068, 2011.

[21] A. Magi, L. Tattini, T. Pippucci, F. Torricelli, and M. Benelli. Read count approach for dna copy number variants detection. *Bioinformatics*, 28(4):470–478, 2012.

[22] S. A. McCarroll, F. G. Kuruvilla, J. M. Korn, S. Cawley, J. Nemesh, A. Wysoker, M. H. Shapero, P. I. de Bakker, J. B. Maller, A. Kirby, et al. Integrated detection and population-genetic analysis of snps and copy number variation. *Nature genetics*, 40(10):1166–1174, 2008.

[23] P. Medvedev, M. Stanciu, and M. Brudno. Computational methods for discovering structural variation with next-generation sequencing. *Nature methods*, 6:S13–S20, 2009.

[24] R. E. Mills, K. Walter, C. Stewart, R. E. Handsaker, K. Chen, C. Alkan, A. Abyzov, S. C. Yoon, K. Ye, R. K. Cheetham, et al. Mapping copy number variation by population-scale genome sequencing. *Nature*, 470(7332):59–65, 2011.

[25] J. I. Naus. The distribution of the size of the maximum cluster of points on a line. *Journal of the American Statistical Association*, 60(310):532–538, 1965.

[26] J. I. Naus. Approximations for distributions of scan statistics. *Journal of the American Statistical Association*, 77(377):177–183, 1982.

[27] D. B. Neill and A. W. Moore. Anomalous spatial cluster detection. *Proceedings of the KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, 2005.

[28] D. B. Neill, A. W. Moore, F. Pereira, and T. M. Mitchell. Detecting significant multidimensional spatial clusters. In *Advances in Neural Information Processing Systems*, pages 969–976, 2004.

[29] S. Pabinger, A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M. R. Speicher, J. Zschocke, and Z. Trajanoski. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in bioinformatics*, 15(2):256–278, 2014.

[30] P. A. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proc. 18th Int. Conf. on Intelligent Systems for Mol. Biol.*, pages 269–278, 2000.

[31] N. Rieber, M. Zapatka, B. Lasitschka, D. Jones, P. Northcott, B. Hutter, N. Jäger, M. Kool, M. Taylor, P. Lichter, et al. Coverage bias and sensitivity of variant calling for four whole-genome sequencing technologies. *PLoS One*, 8(6):e66621, 2013.

[32] S. M. Teo, Y. Pawitan, C. S. Ku, K. S. Chia, and A. Salim. Statistical challenges associated with detecting copy number variations with next-generation sequencing. *Bioinformatics*, 28(21):2711–2718, 2012.

[33] T. J. Treangen and S. L. Salzberg. Repetitive dna and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 13(1):36–46, 2012.

[34] R. Xi, A. G. Hadjipanayis, L. J. Luquette, T.-M. Kim, E. Lee, J. Zhang, M. D. Johnson, D. M. Muzny, D. A. Wheeler, R. A. Gibbs, et al. Copy number variation detection in whole-genome sequencing data using the bayesian information criterion. *Proceedings of the National Academy of Sciences*, 108(46):E1128–E1136, 2011.

[35] R. Xi, T.-M. Kim, and P. J. Park. Detecting structural variations in the human genome using next generation sequencing. *Briefings in functional genomics*, page elq025, 2011.

[36] C. Xie and M. T. Tammi. Cnv-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC bioinformatics*, 10(1):80, 2009.

[37] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome research*, 19(9):1586–1592, 2009.