CLEVER: Clique-Enumerating Variant Finder

Tobias Marschall[1*], Ivan Costa[2*], Stefan Canzar[1], Markus Bauer[3],
Gunnar Klau[1], Alexander Schliep[4], Alexander Schönhuth[1†]

[1] Centrum Wiskunde & Informatica, Amsterdam, Netherlands
[2] Federal University of Pernambuco, Recife, Brazil
[3] Illumina Inc., Cambridge, UK
[4] Rutgers, The State University of New Jersey, Piscataway, NJ, USA
[*] Joint first authorship
[†] Corresponding author

alexander.schoenhuth@cwi.nl

March 6, 2012

Abstract

Next-generation sequencing techniques have for the first time facilitated a large scale analysis of human genetic variation. However, despite the advances in sequencing speeds, achieved at ever lower costs, the computational discovery of structural variants is not yet standard. It is likely that a considerable amount of variants have remained undiscovered in many sequenced individuals.

Here we present a novel internal segment size based approach, which organizes *all*, including also concordant reads into a *read alignment graph* where max-cliques represent maximal contradiction-free groups of alignments. A specifically engineered algorithm then enumerates all max-cliques and statistically evaluates them for their potential to reflect insertions or deletions (indels). We achieve highly favorable performance rates in particular on indels of sizes 30–99 bp. Beyond superior recall and precision, we predict nearly 25% of the annotations as *the only tool* whereas none of the other approaches makes more than 6% such unique and correct predictions. We achieve favorable performance rates also on larger indels (≥ 100 bp) and predict a non-negligible amount of correct, but so far undiscovered variants here as well. On very short indels ($10 - 29$ bp) we outperform all prior insert size approaches, while our unique predictions favorably complement the predictions of the split-read aligner considered.

Our implementation is available from `http://clever-sv.googlecode.com` as open source software under the terms of the GNU General Public License.

Keywords: Structural Variant Detection, Insertions and Deletions, Internal Segment Size, Read Alignment Graph, Maximal Cliques, Algorithm Engineering, Statistical Hypothesis Testing

1 Introduction

The International HapMap Consortium [2005] and The 1000 Genomes Project Consortium [2010] have, through globally concerted efforts, provided the first systematic view into the gamut and prevalence of human genetic variation, including also larger genomic rearrangements. A staggering 8% of the general human population have copy number variants (CNV) affecting regions larger than 500kbp [Itsara *et al.* 2009]. The technology enabling this advance was next-generation sequencing and the reduction in costs and increases of sequencing speeds it brought along [Bentley *et al.* 2008; AppliedBiosystems 2009; Eid *et al.* 2009]. The analysis of structural variation however has not kept up with the advances in sequencing insofar as genotyping of human structural variation has not yet become a routine procedure [Alkan *et al.* 2011]. Indeed it is likely that existing data sets contain structural variations indiscoverable by current methods. These limitations are likewise an obstacle to personalized genomics.

In the following we will not distinguish between the biological terms indel, CNV or structural variant, neither will we distinguish between novel sequence insertions and duplications, or consider balanced genomic re-arrangements. Instead we will talk only about *deletions or insertions (indels)*. In particular the discovery of smaller indels of sizes 1 to 10,000 base pairs (bp), is still a challenge [Alkan *et al.* 2011; Mills *et al.* 2011], even in non-repetitive areas of the genome. That the majority of structural variants resides in repetitive areas complicates the problem further due to the ambiguities in read mapping to such regions.

Categorization of our and prior work. In the following, a *(paired-end) read* is a fragment of DNA both ends of which have been sequenced. We refer to the sequenced ends of the read as *(read) ends* and to the unsequenced part of the fragment between the two ends as *internal segment* or *insert*. An *alignment* A of a paired-end read is a pair of alignments of both ends. We say that a read has been *multiply mapped* if it aligns at several locations in the reference genome and *uniquely mapped* in case of only one alignment. Existing approaches for structural variant discovery can be classified into three broad classes: first, those based on the read alignment coverage, that is, the number of read ends mapping to a location [Campbell *et al.* 2008; Chiang *et al.* 2009; Alkan *et al.* 2009; Sudmant *et al.* 2010; Yoon *et al.* 2009; Abyzov *et al.* 2011], second, those analyzing the paired-end read internal segment size [Korbel *et al.* 2009; Hormozdiari *et al.* 2009; Chen *et al.* 2009a; Lee *et al.* 2009; Sindi *et al.* 2009; Quinlan *et al.* 2010], and third, split-read alignments [Mills *et al.* 2006; Ye *et al.* 2009]. See [Medvedev *et al.* 2009] as well as [Alkan *et al.* 2011] for reviews, and in the latter Figure 2 for illustrations of the basic ideas behind the different classes of approaches. A major difference is that the first two classes align short reads by standard read mappers, such as BWA [Li and Durbin 2009], Mr and MrsFast [Alkan *et al.* 2009; Hach *et al.* 2010] and Bowtie [Langmead *et al.* 2009], to only name a few most popular and recent ones, see [Li and Homer 2010] for a review. Split-read aligners however compute custom alignments which span breakpoints of putative insertions and deletions. They usually have advantages over insert size based approaches on smaller indels while performing worse in predicting larger indels.

It is well justified to assume that the length of the internal segment follows a normal distribution with machine- and protocol-specific mean μ and standard deviation σ. A common definition then are *concordant and discordant alignments*: an alignment with interval length $I(A)$ (see Figure 1) is concordant iff $|I(A) - \mu| \leq K\sigma$ and discordant otherwise. The constant K can vary among the different approaches. A *concordant read* is defined to concordantly align with the reference genome, that is, it should give rise to at least one concordant alignment.

With only one exception [Lee *et al.* 2009, MoDIL], all prior approaches discard concordant reads. In this paper, we present CLEVER, a novel insert size based approach that takes *all, also concordant reads* into consideration. While a single discordant read is significantly likely to testify the existence of a structural variant, a single concordant read only conveys weak variant signals if any. Ensembles of *consistent concordant alignments* however can provide significant evidence of usually smaller variants. The major

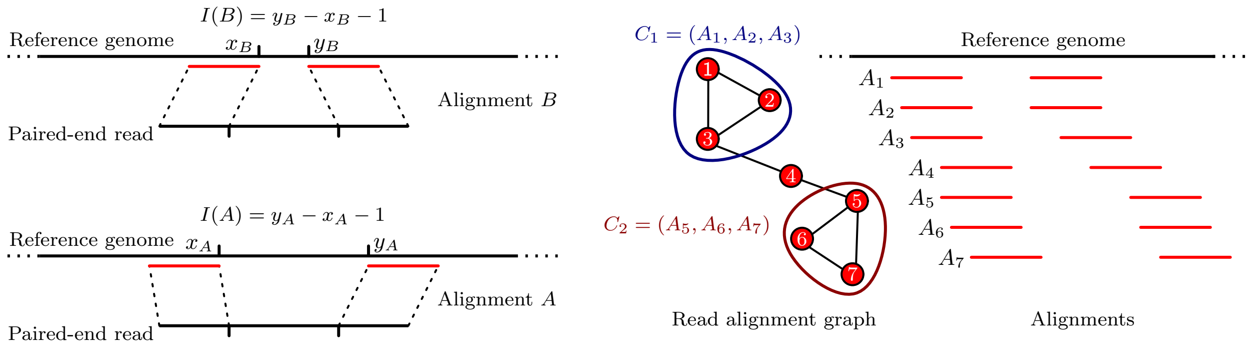


Figure 1: Left part: two read alignments. Assuming $I(A) > \mu > I(B)$ where μ is the mean of the true insert size distribution, alignment A is likely to indicate a deletion while alignment B may indicate an insertion. Right part: Read alignment graph for seven closely located read alignments. Note that $1/3(I(A_5)+I(A_6)+I(A_7)) > 1/3(I(A_1)+I(A_2)+I(A_3))$. Assuming that all alignments have equal weight, C_2 is more likely to indicate a deletion than C_1 through a hypothesis test of the type (3),(2). Note that we have not marked cliques (A_3, A_4) and (A_4, A_5). See Fig. 2 for definition of edges.

motivation of this study is to address this systematically in order to not miss any significant variant signal among concordant reads. We do this to increase recall in particular for smaller indels and detect variants that cannot be discovered by existing tools.

We employ a statistical framework, which addresses deviations in insert size, alignment quality, multiply mapped reads and coverage fluctuations in a principled manner. As a result, our approach outperforms all prior insert size approaches on both simulated and real data and also compares favorably with a state-of-the-art split-read aligner. Beyond its favorable results, our tool predicts more than 20% of the correctly annotated indels of sizes $30-99$ bp *as the only tool*—whereas none of the previous approaches makes more than 6% unique and correct predictions. On very small indels ($10-29$ bp), CLEVER's still non-negligible amount of unique predictions complement the predictions of the split-read aligner considered [Ye *et al.* 2009, PINDEL].

Moreover, we need approximately 8 hours on a single CPU for a 30x coverage whole-genome dataset with approximately 1 billion reads, which compares favorably with estimated 7,000 CPU hours needed by MoDIL, the only method that also takes into consideration all reads.

1.1 Approach and Related Work

1.1.1 Graph-Based Framework

Our approach is based on organizing all read alignments into a read alignment graph whose nodes are the alignments and edges reflect that the reads behind two overlapping alignments are, in rigorous statistical terms, likely to stem from the same allele. Accordingly, maximal cliques (max-cliques) reflect maximal consistent groups of alignments that are likely to stem from the same location in a donor allele. Since we do not discard alignments, the number of nodes in our read alignment graph is large: more than 10^9 nodes in the instances considered here. We determine all max-cliques in this graph by means of a specifically engineered, fast algorithmic procedure.

The idea to group alignments into location-specific, consistent ensembles, such as max-cliques here, is not new. In fact, it has been employed in the vast majority of previous insert size based approaches. We briefly discuss related concepts of the three most closely related approaches, by Hormozdiari *et al.* [2009, VariationHunter (VH)], Sindi *et al.* [2009, GASV] and Quinlan *et al.* [2010, HYDRA]. Although not framing it in rigorous statistical terms, HYDRA is based on precisely the same concept of a max-clique as in our approach. After constructing the read alignment graph from discordant reads alone, they employ a heuristic algorithm for mining max-cliques. Since no theoretical guarantee is given, it remains unclear whether HYDRA enumerates them all. The definition of a 'valid cluster' in VH [Hormozdiari *et al.* 2009] differs in a subtle, but decisive aspect from our cliques: in the read alignment graph, VH draws edges already if only (6) from below is satisfied—(7) however may be violated. Consequently, each of our max-cliques forms a valid cluster, but the opposite is not necessarily true. The reduction in assumptions however allows VH to compute valid clusters as max-cliques in interval graphs, in a nested fashion, which yields a polynomial-runtime algorithm. Sindi *et al.* [2009, GASV] use a geometrically motivated definition which allows application of an efficient plane-sweep style algorithm, which has its roots in computational geometry. A closer look reveals that each geometric arrangement of alignments inferred by GASV constitutes a max-clique in our sense, but not necessarily vice versa, even if a max-clique is formed by only discordant read alignments. We recall that GASV, HYDRA and VH do not consider concordant read data hence consider read alignment graphs of much reduced sizes.

Finding maximal cliques is $\mathcal{NP}$-hard in general graphs. Based on the idea that the read alignment graph we consider still largely resembles an interval graph, we provide a specifically engineered routine that computes and tests all max-cliques in reasonable time—about 1h on a current 8 core machine for a whole human genome sequenced to 30x coverage—despite that we do not discard any read.

1.1.2 Significance Evaluation

Commonly Concordant and Discordant Reads. Testing whether $|I(A) - \mu| \leq K \cdot \sigma$, to determine whether a single alignment is concordant, is equivalent to performing a Z-test at significance level $p_K := 1 - \Phi(K)$ where Φ is the standard normal distribution function. However, when determining whether m consistent alignments (such as a clique of size m) with mean interval length $\bar{I}$ are *commonly concordant*, a Z-test for a sample of size m is required, which translates to

$$1 - \Phi(\sqrt{m} \cdot \frac{|\bar{I} - \mu|}{\sigma}) \geq p_K \;\Leftrightarrow\; \sqrt{m} \cdot |\bar{I} - \mu| \leq K \cdot \sigma. \tag{1}$$

Due to the factor $\sqrt{m}$, already smaller deviations $|\bar{I} - \mu|$ turn out to render the alignments *commonly discordant*. In our approach, we rigorously expand on this idea—in a rough description, each max-clique undergoes a (1)-like hypothesis test.

Multiply Mapped Reads. We also address multiply mapped reads in a statistically principled way. While we approach the idea of not "overusing" multiply mapped reads in an essentially different fashion, our routine serves analogous purposes as the set-cover routines inherent to VH and HYDRA. The difference is that we only try to keep control of read mapping ambiguity, but do not necessarily aim at resolving it, which we see as an interesting direction for future work.

Here, patterned after [Li *et al.* 2008], we compute probabilities for alignments which reflect that they indicate the correct placement of their read. In case of a max-clique consisting of alignments $A_1, ..., A_n$ (all from different reads) with probabilities $p_1, ..., p_n$, let $A_J, J \subset \{1, ..., n\}$ be the event that precisely the alignments $A_j, j \in J$ are correct. We compute $\mathbf{P}(A_J) = \prod_{j \in J} p_j \prod_{j \notin J}(1 - p_j)$. Let H_0 be the null hypothesis of that the allele in question—we recall that max-cliques just represent groups of alignments likely to be from the same allele—coincides with the reference genome. In correspondence to (1), we compute ($\bar{I}_J = \frac{1}{\sum_{j \in J} p_j} \sum_{j \in J} p_j I(A_j)$)

$$\mathbf{P}(H_o \mid A_J) := 1 - \Phi(\sqrt{|J|} \frac{|\bar{I}_J - \mu|}{\sigma}) \tag{2}$$

as the tail probability of that the null hypothesis holds true given that precisely the alignments $A_j, j \in J$ are correct and further

$$\mathbf{P}(H_0) = \mathbf{P}(H_0 \mid A_1, ..., A_n) = \sum_{J \subset \{1,...,n\}} \mathbf{P}(A_J)\mathbf{P}(H_0 \mid A_J) \tag{3}$$

as the probability [note that $\sum_{J \subset \{1,...,n\}} P(A_J) = 1$] that max-clique $A_1, ..., A_m$ does *not* support an indel variant. We further correct $\mathbf{P}(H_o)$ with a *local Bonferroni factor* to adjust for coverage-mediated fluctuations in the number of implicitly performed tests. If the corrected $\mathbf{P}(H_0)$ is significantly small, it is likely that (at least) one allele in the donor is affected by an indel at that location. See Methods for precise details. In a last step, we apply the Benjamini-Hochberg procedure to correct for multiple hypothesis testing overall. Note that none of the previous approaches addresses to correct for multiple hypothesis testing, although they either explicitly (e.g. [Lee *et al.* 2009; Chen *et al.* 2009b]) or implicitly (e.g. [Hormozdiari *et al.* 2009; Korbel *et al.* 2009; Quinlan *et al.* 2010]) perform multiple hypothesis tests.

Lee *et al.* [2009]; Chen *et al.* [2009b] also employ statistical significance considerations. While Lee *et al.* [2009], after clustering, use Kolmogorov-Smirnov tests in combination with bimodality assumptions, Chen *et al.* [2009b] measure both deviations from Poisson-distribution based assumptions (BreakdancerMax) and use Kolmogorov-Smirnov (BreakdancerMin) tests to discover copy number changes.

2 Methods

2.1 Notations, Definitions and Background

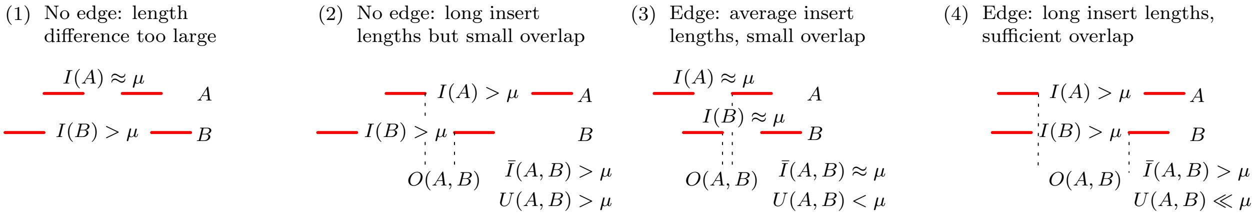
Reads and Read Alignments. Let $\mathcal{R}$ be a set of paired-end reads, stemming from a *donor (genome)* which have been aligned against the *reference (genome)*. We write A for a paired-end alignment, that is a pair of alignments of the two ends of a read (see Fig. 1) and $\mathcal{A}(R)$ for the set of correctly oriented alignments which belong to read R. We neglect incorrectly oriented alignments, with no measurable effects on the discovery of copy number variants. We write $\mathcal{A} = \cup_R \mathcal{A}(R)$ for the set of all alignments we consider. We assume that $|\mathcal{A}(R)| \geq 1$ that is each read we consider gives rise to a well-oriented alignment; note that $|\mathcal{A}(R)| > 1$ translates to a multiply mapped read. As pointed out in the Introduction, we do not discard any reads.

We write x_A for the rightmost position of the left end and y_A for the leftmost position of the right end. We write $[x_A + 1, y_A - 1]$ and call this the *interval* of alignment A (in slight abuse of notation: intervals here only contains integers) and $I(A) := y_A - x_A - 1$ for the *(alignment) interval length*. When referring to alignment intervals, we sometimes call x_A, y_A the left and right *endpoint*. See Figure 1 for illustrations.

Internal Segment Size Statistics. We write $I(R)$ for the true (but unknown) internal segment size of paired-end read R which is the (unknown) length of the entire read R minus the (known) lengths of its two sequenced ends. It is reasonable to assume that $I(R)$ is normally distributed where

$$I(R) \sim \mathcal{N}_{(\mu,\sigma)} \tag{4}$$

for appropriately chosen mean μ and standard deviation σ [Li *et al.* 2008; Li and Durbin 2009; Hormozdiari *et al.* 2009; Lee *et al.* 2009]. Estimation of mean μ and standard deviation σ poses the challenge that the empirical statistics on alignment length, further denoted as $\mathbf{P}_{\text{Emp}}$ are fat-tailed, due to already reflecting structural variation between donor and reference. Here, we rely on robust estimation routines, as implemented by BWA [Li and Durbin 2009].


(1) No edge: length difference too large

(2) No edge: long insert lengths but small overlap

(3) Edge: average insert lengths, small overlap

(4) Edge: long insert lengths, sufficient overlap

Figure 2: Four scenarios of two overlapping alignment pairs A and B. In the *read alignment graph*, two alignments are connected by an *edge* if they are compatible, that is, they support the same phenomenon. (1) Alignment A has an insert length about the expected insert length μ, suggesting that there is no variation present but alignment B has an insert length much larger than μ suggesting a deletion. Hence, A and B are not compatible. (2) Both alignments have similar insert lengths larger than μ, both suggesting a deletion of size $I(A) - \mu \approx I(B) - \mu$, but the overlap $O(A, B)$ is too small to harbor a deletion of this size. Thus, they are incompatible. (3) Both alignments do not suggest any variation and are therefore compatible. (4) Similar to Case (2), but now the overlap is large enough to contain the putative deletion.

Alignment Scores and Probabilities. As described by Li *et al.* [2008], we determine $\log_{10} \mathbf{P}_{\text{Ph}}(A) := -\sum_j Q_j/10$ where j runs over all mismatches in both read ends and Q_j is the phred score for position j, that is $10^{-(Q_j/10)}$ is the probability that the nucleotide at position j reflects a sequencing error. Hence $\mathbf{P}_{\text{Ph}}(A)$ is the probability that the substitutions in alignment A are due to sequencing errors. The greater $\mathbf{P}_{\text{Ph}}(A)$ the more likely that A is correct so $\mathbf{P}_{\text{Ph}}(A)$ serves as a statistical quality assessment of A. Note that to neglect SNP rates and indels reflects common practice [Li *et al.* 2008; Li and Durbin 2009], which is justified by that in Illumina reads substitution error rates are higher than SNP rates, indel sequencing error rates and DIP (deletion/insertion polymorphism) rates by orders of magnitude [Bravo and Irizarry 2010; Albers *et al.* 2011].

Patterned according to Li *et al.* [2008]; Li and Durbin [2009], we integrate the empirical interval length distribution $\mathbf{P}_{\text{Emp}}(I(A))$ into an overall score $S_0(A) := \mathbf{P}_{\text{Ph}}(A) \cdot \mathbf{P}_{\text{Emp}}(I(A))$ and obtain as the probability that A is the correct alignment for its read, by application of Bayes' formula

$$\mathbf{P}_0(A) = \frac{S_0(A)}{\sum_{\tilde{A} \in \mathcal{A}(R)} S_0(\tilde{A})}. \tag{5}$$

The Read Alignment Graph. We arrange all scored read alignments $\mathcal{A}$ in form of an undirected, weighted graph $G = (\mathcal{A}, E, w)$. Since we identify nodes with read alignments from $\mathcal{A}$, we use these terms interchangeably. We draw an edge between alignments $A, B \in \mathcal{A}$ if we cannot reject the hypothesis that, in case they are both correct, their reads can stem from the same allele. See the subsequent paragraph for details. The weight function on the nodes $w : \mathcal{A} \to [0, 1]$ is defined by $w(A) := \mathbf{P}_0(A)$. We further label nodes by $r : \mathcal{A} \to \{1, ..., N\}$ where $r(A) = n$ iff $A \in \mathcal{A}(R_n)$ that is alignment A is due to read R_n.

As usual, we write $\delta(A) := |\{B \in \mathcal{A} \mid (A, B) \in E\}|$ for the *degree* of node A. A *clique* $\mathcal{C} \subset \mathcal{A}$ is defined as a subset of mutually connected nodes, that is, $(A, B) \in E$ for all $A, B \in \mathcal{C}$. A *maximal clique* $\mathcal{C}$ is a clique such that for every node $A \in \mathcal{A} \setminus \mathcal{C}$ there is $B \in \mathcal{C} : (A, B) \notin E$. Note that by our definition of edges, a clique is a group of alignments which can be jointly assumed to be associated with the same allele, or, in other words, to jointly support the same local phenomenon in the donor genome. Maximal cliques are obviously particularly interesting: while all alignments in the clique are likely to support the same local phenomenon, joining any other *overlapping* alignment may lead to conflicts. Hence a maximal clique is likely to commonly support a certain phenomenon in the donor genome.

Edge Computation. See Figure 2 for illustrations of the following. Let A, B be two alignments. We define:

- $\Delta(A, B) := |I(A) - I(B)|$ is the absolute difference of interval length.

- $O(A, B) := \min(y_A, y_B) - \max(x_A, x_B) - 1$ where in case of $O(A, B) \geq 0$ we refer to all positions between $\max(x_A, x_B)$ and $\min(y_A, y_B)$ as their *common interval*. Note that $O(A, B) < 0$ reflects that A and B's internal segments cannot commonly harbor a variant.

- $\bar{I}(A, B) := (I(A) + I(B))/2$ is the *mean of A and B's interval lengths*.

- $U(A, B) := \bar{I}(A, B) - O(A, B)$, is the difference of mean interval length and overlap. To motivate this quantity, note that, in case A and B overlap (hence length of common interval $O(A, B) > 0$) and are from the same allele, a deletion at that location can only happen to take place in their common interval. If $U(A, B)$ is large then $\bar{I}(A, B)$ significantly deviates from μ and the common interval is not large enough to explain this by a large enough deletion. Hence it is unlikely A, B are from the same allele.

Let X be $\mathcal{N}_{(0,1)}$-distributed and, as above, μ, σ be mean and variance of the insert size distribution. We draw an edge between alignments A, B in the read alignment graph iff the reads of A and B are different, $O(A, B) \geq 0$ and

$$\mathbf{P}(|X| \geq \frac{1}{\sqrt{2}} \frac{\Delta(A, B)}{\sigma}) \leq 0.05 \quad \text{and} \tag{6}$$

$$\mathbf{P}(X \geq \sqrt{2} \frac{(U(A, B) - \mu)}{\sigma}) \leq 0.05. \tag{7}$$

(6) is a two-sided two sample Z-test to measure *statistically compatible insert size*. (7) reflects a two-sided one-sample Z-test for *statistically consistent overlap* [Wasserman 2004]. If two alignments A, B with $O(A, B) \geq 0$ pass these tests, we have no reason to reject the hypothesis that the alignments are from the same allele so we draw an edge. We recall that similar ideas have been presented by Quinlan *et al.* [2010, HYDRA].

2.2 CLEVER: Algorithmic Workflow

1. Enumerating Maximal Cliques: We compute all *maximal cliques* in the read alignment graph.

2. We assign two p-values $p_D(\mathcal{C}), p_I(\mathcal{C})$ to each maximal clique $\mathcal{C}$ which are the probabilities that the alignments participating in $\mathcal{C}$ do not commonly support a deletion or insertion. So the lower $p_D(\mathcal{C})$ resp. $p_I(\mathcal{C})$, the more likely that $\mathcal{C}$ supports a deletion resp. insertion.

3. For the thus computed p-value, we control the false discovery rate at 10 % by applying the standard Benjamini-Hochberg procedure separately for insertions and deletions. All cliques remaining after this step are deemed *significant* and processed further.

4. Determining Parameters: We parametrize deletions D by their left breakpoint D_B and their length D_L, which denotes that reference nucleotides of positions $D_B, ..., D_B + D_L - 1$ are missing in the donor. We parametrize insertions I by their breakpoint I_B and their length I_L such that before position I_B in the reference there has been a sequence of length L inserted in the donor. Depending on whether $\mathcal{C}$ represents a deletion or insertion, we determine $[w(\mathcal{C}) := \sum_{A \in \mathcal{C}} w(A)]$

$$\frac{1}{w(\mathcal{C})} \sum_{A \in \mathcal{C}} w(A)(I(A) - \mu) \quad \text{resp.} \quad \frac{1}{w(\mathcal{C})} \sum_{A \in \mathcal{C}} w(A)(\mu - I(A)) \tag{8}$$

as the length D_L of the deletion resp. I_L of the insertion. We determine breakpoints D_B or I_B such that the predicted deletion or insertion sits right in the middle of the intersection of all internal segments of alignments in $\mathcal{C}$.

Enumerating Maximal Cliques: Algorithm Engineering. Assuming an *interval graph* is given as a set of intervals on the real line, its maximal cliques can be found efficiently in time linear in the size of the graph [Fishburn 1985]. This algorithm is applied in [Hormozdiari *et al.* 2010] to find maximal valid clusters for copy events. In [Hormozdiari *et al.* 2009], valid clusters for insertion and deletion events are characterized by overlapping intervals of similar size. Both requirements can be modeled naturally by intervals that must intersect and thus maximal cliques can be determined again by applying the above mentioned algorithm twice.

We identify nodes of the read alignment graph by the intervals of the corresponding alignments. We first sort the $2m$ endpoints of these intervals, $m := |\mathcal{A}|$, in ascending order of their values. We then scan this list from left to right. We maintain a set of *active* cliques that could potentially be extended by a subsequent interval, which initially is empty. If the current element ℓ of the list is a left endpoint, we extend the set of active cliques according to the following rules. For the sake of simplicity, let us assume that a unique interval ends at ℓ, corresponding to a vertex A in the read alignment graph G. Let $N(A)$ be the open neighborhood of A. If $\mathcal{C} \cap N(A) = \emptyset$ for all active cliques $\mathcal{C}$, add a singleton clique $\{A\}$ to the set of active cliques. Otherwise, for each active clique $\mathcal{C}$,

- if $\mathcal{C} \cap N(A) = \mathcal{C}$, then $\mathcal{C} := \mathcal{C} \cup \{A\}$, otherwise
- if $\mathcal{C} \cap N(A) \neq \emptyset$, add $(\mathcal{C} \cap N(A)) \cup \{A\}$ to the set of active cliques.

Finally, duplicates and cliques that are subsets of others are removed.

If the current element ℓ of the list is a right endpoint, we output all cliques that contain at least one interval ending at ℓ. These cliques go out of scope and are thus maximal. We remove intervals ending at ℓ from active cliques. Cliques that become empty are removed from the set of active cliques.

Concerning the complexity of the algorithm, the sorting step takes $\mathcal{O}(m \log m)$. Implemented in a naïve way, the intersection of the neighborhood of the current vertex with all active cliques can be determined by iterating over all vertices contained in active cliques, which we bound to be at most $k := 500$, see below. This gives a total running time of $\mathcal{O}(m(\log m + k + c) + s)$, where c is the maximum size of the set of active cliques and s is the size of the output.

Note that the read alignment graph is never constructed explicitly. Rather, we compute the edges on demand. Since the computation of $N(A) \cap \mathcal{C}$ for all active cliques $\mathcal{C}$ is the key step of our algorithm, we use a binary search tree in combination with bit-parallel operations that considerably improve the perfomance of our algorithm, although they do not have any effect on its worst case analysis.

The key idea leading to a practically fast algorithm is to represent each active clique as a *bitvector*, where each bit indicates whether a particular node is part of the clique or not. We keep all nodes from active cliques in a binary search tree sorted by their segment length, such that vertices whose alignment satisfy condition (6) can be found efficiently. We test each one of these candidate vertices for condition (7) to identify the neighborhood of the current vertex u. With our representation of the current cliques as bitvectors, we can compute all the intersections with $N(u)$ by a bitparallel boolean operation.

When nodes become inactive, a reorganization of all bitarrays is required and doing this in each iteration would cancel the benefits of the fast bitparallel operations. To have a good trade-off between not-too-frequent memory reorganizations and not-too-large active sets of nodes, we employ the following strategy. We start with a bitvector capacity equaling the machine word size (usually 64 bits) and reserve this amount of memory for each clique (although fewer nodes are active in the beginning). Whenever the number of active nodes reaches the capacity, we reorganize the data structure. That is, we discard all now inactive nodes, set the new capacity to twice the size of the set of now active nodes, and repack all bitvectors.

We bound local alignment coverage by first removing alignments of interval length $\geq 50,000$, due to that discovery of deletions of that size is considered rather easy [Alkan *et al.* 2011]. We further remove alignments of weight $\mathbf{P}_0(A) < 1/625$ if necessary, motivated by that we allow at most 25 alignments per read end that is $25^2 = 625$ alignments per paired-end read (see Results). We found that these restrictions result in at most ≈ 500 alignments also in heavily repetitive areas.

P-Values for Cliques. We proceed as sketched in the discussion in the Introduction, surrounding (2) and (3). Let $\mathcal{C}$ be a maximal clique in the read alignment graph and let $w(\mathcal{C}) := \sum_{A\in\mathcal{C}} w(A) = \sum_{A\in\mathcal{C}} \mathbf{P}_0(A)$ be the *the weight of the clique*. Let $\bar{I}(\mathcal{C}) := \frac{1}{w(\mathcal{C})} \cdot \sum_{A\in\mathcal{C}} w(A) \cdot I(A)$ be the *weighted mean of alignment interval length* of the clique. Let Φ be the standard normal distribution function. Let $\rho(\mathcal{C})$ be the number of alignments which are at the genomic location of the clique. For example, in Figure 1, $\rho(C_1) = \rho(C_2) = 7$ is just the number of alignments which overlap with one another at this position of the reference. We compute

$$p(\mathcal{C})_D := 2^{\rho(\mathcal{C})} \sum_{J\subset\mathcal{C}} \mathbf{P}(A_J)[1 - \Phi(\sqrt{|J|}\frac{\bar{I}(\mathcal{C}) - \mu}{\sigma})] \tag{9}$$

$$p(\mathcal{C})_I := 2^{\rho(\mathcal{C})} \sum_{J\subset\mathcal{C}} \mathbf{P}(A_J)[\Phi(\sqrt{|J|}\frac{\bar{I}(\mathcal{C}) - \mu}{\sigma})] \tag{10}$$

just as in (3),(2) with the difference that we distinguish between cliques which give rise to deletions and insertions. $2^{\rho(\mathcal{C})}$ is the number of subsets of alignments one can test at this location, that is the virtual number of tests which we perform, so multiplying by $2^{\rho(\mathcal{C})}$ is a Bonferroni-like correction. This correction accounts for coverage fluctuations: note that one expects max-cliques to be larger at locations of higher coverage.

If $p(\mathcal{C})_D$ is significantly small then $\bar{I}(\mathcal{C})$ is significantly large, hence the alignments in $\mathcal{C}$ are deemed to commonly support a deletion. Analogously, if $p(\mathcal{C})_I$ is significantly small, then $\mathcal{C}$ is supposed to support an insertion.

We need to address one last caveat: the sum $\sum_{J\subset\mathcal{C}}$ consists of exponentially many summands. In order to ensure fast enough runtime, we opted for an efficient approximation scheme, which, for example in case of only uniquely mapped alignments yields exact values.

In the following, we describe how to compute such reasonable approximations $\mathbf{P}^*(\mathcal{C})$ for $\mathbf{P}(H_0 \mid \mathcal{C})$ in polynomial time. Due to that we would like to ensure to keep false discovery rate under control when correcting for multiple testing, $\mathbf{P}^*(\mathcal{C})$ should be an *upper bound* for $\mathbf{P}(H_0 \mid \mathcal{C})$.

Let $\mathcal{A}$ be a set of alignments and $w_{max}(\mathcal{A}) := \max\{w(A) \mid A \in \mathcal{A}\}$ resp. $w_{min}(\mathcal{A}) := \min\{w(A) \mid A \in \mathcal{A}\}$ be the maximum resp. minimum weight of an alignment $A \in \mathcal{A}$. As approximation scheme for (9), we first determine

$$w_{max}(\mathcal{C}) := \max\{w(A) \mid A \in \mathcal{C}\} \tag{11}$$

for the clique $\mathcal{C}$ in question and further (L for large, S for small weight)

$$\mathcal{C}_L := \{A \in \mathcal{C} \mid w(A) \geq \frac{1}{2} w_{max}(\mathcal{C})\} \quad \text{and} \tag{12}$$

$$\mathcal{C}_S := \{A \in \mathcal{C} \mid w(A) < \frac{1}{2} w_{max}(\mathcal{C})\}. \tag{13}$$

Let further $\mathcal{C}_k \subset \mathcal{C}$ be the k "most concordant" alignments in clique $\mathcal{C}$, that is, $A \in \mathcal{C}_k$ iff

$$|I(A) - \mu| \leq |I(B) - \mu| \tag{14}$$

for at least $|\mathcal{C}| - k$ alignments $B \in \mathcal{C}$. Let $\mathbf{P}(H_0 \mid \mathcal{C}_k)$ be the probability that the null hypothesis of no variant holds true given that precisely the k most concordant alignments $\mathcal{C}_k$ are correct. As in the main text, we assume that $\mathcal{C} = \{A_1, ..., A_n\}$ consists of n alignments and we write $A_J, J \subset \mathbb{N}_n := \{1, ..., n\}$ for the event that precisely the alignments $A_j, j \in J$ are correct. By definition of $\mathcal{C}_k$, for each $J \subset \mathbb{N}_n$ with cardinality $|J| = k$

$$\mathbf{P}(H_0 \mid A_J) \leq \mathbf{P}(H_0 \mid \mathcal{C}_k). \tag{15}$$

Let further $J \subset \mathbb{N}_n$ be of cardinality $|J| = k$ such that

$$|\{A_j, j \in J\} \cap \mathcal{C}_L| = l \tag{16}$$

that is l alignments of the $A_J, j \in J$ are from $\mathcal{C}_L$ which translates to that they have comparatively large weight $w(A_j)$. If $0 \leq l \leq |\mathcal{C}_L|$ there are $\binom{|\mathcal{C}_L|}{l} \cdot \binom{|\mathcal{C}_S|}{k-l}$ such subsets J. For each such subset, we compute

$$\mathbf{P}(A_J) \leq w_{k,l}(\mathcal{C}) := w_{max}(\mathcal{C}_L)^l w_{max}(\mathcal{C}_S)^{k-l} \cdot (1 - w_{min}(\mathcal{C}_L))^{|\mathcal{C}_L|-l}(1 - w_{min}(\mathcal{C}_S))^{|\mathcal{C}_S|-(k-l)}. \tag{17}$$

We compute $[\binom{m_1}{m_2} := 0, m_2 > m_1]$

$$\sum_{J \subset \mathbb{N}_n, |J|=k} \mathbf{P}(A_J)\mathbf{P}(H_0 \mid A_J) \overset{(15),(17)}{\leq} \mathbf{P}(H_0 \mid \mathcal{C}_k) \cdot \sum_{l=0}^{|\mathcal{C}_L|} w_{k,l} \cdot \binom{|\mathcal{C}_L|}{l} \cdot \binom{|\mathcal{C}_S|}{k-l} \tag{18}$$

which overall amounts to

$$\begin{aligned} \mathbf{P}(H_0 \mid \mathcal{C}) &= \sum_{J \subset \mathbb{N}_n} \mathbf{P}(A_J)\mathbf{P}(H_0 \mid A_J) = \mathbf{P}(A_\emptyset) + \mathbf{P}(A_{\mathbb{N}_n}) + \sum_{k=1}^{n-1} \sum_{J \subset \mathbb{N}_n, |J|=k} \mathbf{P}(A_J)\mathbf{P}(H_0 \mid A_J) \\ &\overset{(18)}{\leq} \prod_{j=1}^{n}(1 - w(A_j)) + \prod_{j=1}^{n} w(A_j) + \sum_{k=1}^{n-1} \mathbf{P}(H_0 \mid \mathcal{C}_k) \sum_{l=0}^{|\mathcal{C}_L|} w_{k,l} \cdot \binom{|\mathcal{C}_L|}{l} \cdot \binom{|\mathcal{C}_S|}{k-l} =: \mathbf{P}^*(\mathcal{C}) \end{aligned} \tag{19}$$

This upper bound can be computed in polynomial time.

We recall that a motivation for our approximation is that if all alignments $A \in \mathcal{C}$ have equal weight, which most importantly covers the case of only uniquely mapped alignments, the approximation yields the exact value.

3 Results and Discussion

Simulation: Craig Venter Reads. We downloaded the comprehensive set of annotations of both homozygous and heterozygous structural variants (also including inversions and all other balanced re-arrangements) for Craig Venter's genome, as documented by Levy *et al.* [2007] and introduced them into the reference genome, thereby generating two different alleles. If nested effects lead to ambiguous interpretations we opted for an order which respects the overall predicted change in copy number. We used UCSC's SimSeq[1] as read simulator to simulate Illumina paired-end reads with read end length 100 at coverage $15x$ for each of the two alleles which yields $30x$ sequence coverage overall.

Real Data: NA18507. We further were provided with reads of the genome of an individual from the Yoruba in Ibadan, Nigeria by Illumina. Reads were sequenced on a GAIIx and are now publicly available[2]. Read ends are of length 101. Read coverage is $30x$. For benchmarking purposes, we used annotations from [Mills *et al.* 2011, Gen.Res.].

[1] https://github.com/jstjohn/SimSeq

[2] ftp://ftp.sra.ebi.ac.uk/vol1/ERA015/ERA015743/srf/

Reference Genome and Alignments As a reference genome, we used version hg 18, as downloaded from the UCSC Genome Browser. All reads considered were aligned using BWA [Li and Durbin 2009] with the option to allow 25 alignments per read end, which amounts to a maximum of 25^2 alignments per paired-end read. BWA determined mean insert size $\mu \approx 112$ and standard deviation $\sigma \approx 15$ for both simulated and NA18507 reads. Note that re-alignment of discordant reads with a slow, but more precise alignment tool, such as Novoalign[3] can lead to subsequent resolution of much misaligned sequence and therefore has been suggested by variant predictor tools [Quinlan *et al.* 2010]. We are aware that all methods considered would benefit from such (time-consuming) re-alignment of reads.

Experiments. For benchmarking, we considered 5 different state-of-the-art insert size based approaches, 4 of which are applicable for a whole-genome study: GASV [Sindi *et al.* 2009], VariationHunter [Hormozdiari *et al.* 2009, v3.0], Breakdancer [Chen *et al.* 2009a] and HYDRA [Quinlan *et al.* 2010]. We ran MoDIL [Lee *et al.* 2009] only on chromosome 1 of the simulated data which, on our machines required several hundred CPU hours. In contrast, we process chromosome 1 in less than one hour. We also consider the split-read aligner PINDEL [Ye *et al.* 2009]. For all methods, we determined indels, as per their default settings. In case of deletions, we defined true positives (TP) as true deletions, which overlap with a predicted deletion, false positives (FP) are predicted deletions, which do not overlap with true deletions and false negatives (FN) are true deletions not hit by a predicted deletion. As usual, *recall* = TP/(TP+FN) and *precision* = TP/(TP+FP). A predicted insertion was considered a true positive when the distance between predicted and true breakpoint was twenty base pairs or less. We refer to *Exc.* (= exclusive) as the percentage of true annotations which were *exclusively (and correctly)* predicted by the method in question. Since the annotations for the real data set are obviously far from complete, a false positive may in fact point out a missing annotation. Note however that the definition of TP and FN apply correctly also on the real data. We therefore call the ratio TP/(TP+FP) *relative precision (RPr.)*. We put related values in parentheses as they have only limited meaning.

Results. See Table 1 for performance figures. As expected, performance rates greatly depend on the size of the indels.

$10 - 29$ bp: GASV, MoDIL and CLEVER are the only internal segment length based approaches that make any correct predictions, where overall CLEVER performs best. The split-read based approach of PINDEL yields the most favorable results. Indels predicted by CLEVER and GASV, however, complement the predictions of PINDEL since they contribute 8 = 5 + 3 = CLEVER Exc. + GASV Exc. % discoveries which PINDEL misses.

$30 - 49$, $50 - 99$ bp: Here, CLEVER clearly outperforms all other insert size approaches and achieves better recall than PINDEL. Most importantly, CLEVER predicts, on average over these categories, 20% of true variants, which are missed by all prior, both insert size and split-read based approaches. None of the other tools predicts more than 6% such unique true variant predictions.

≥ 100 bp: CLEVER still achieves highly favorable values in all categories, incurring only minor losses in precision with respect to other tools. Again, while at lower rates, CLEVER delivers the largest amounts of correctly discovered indels ($\approx 6\%$) which are missed by all other tools also in that size range; on the real data, CLEVER is the only insert size based tool that makes unique such predictions for deletions larger than 500 bp.

Conclusion. We have presented a novel internal segment size based approach for discovering human copy number variation from paired-end read data, which, in contrast to all previous, whole-genome-applicable approaches, takes all concordant read data into account. We outperform all prior insert size based approaches on indels of all sizes and we outperform the split-read based approach considered on medium-sized ($30 - 99$

[3] http://www.novocraft.com/main/index.php

Table 1: This table shows benchmarking results for simulated (Venter) and real data (NA18507). Performance rates as recall, precision and exclusive predictions (Exc. which are true predictions, uniquely predicted by that tool) are grouped by different indel size ranges. A dash resp. N/A indicates no prediction resp. not applicable (e.g. GASV cannot report insertions) in that category. Since MoDIL is extremely runtime intensive, we only ran it on Venter's chromosome 1. Insertions larger than the internal segment size (≈ 112 here) cannot be detected by insert size based approaches. PINDEL does not detect insertions in that size range either.

Data Set	**Venter Insertions**			**Venter Deletions**			**NA18507 Insertions**			**NA18507 Deletions**		
	Prec.	Rec.	Exc.	Prec.	Rec.	Exc.	(RPr.)	Rec.	Exc.	(RPr.)	Rec.	Exc.
Length Range 10–29 (24 279 true ins., 26 076 true del.)							(4 847 true ins., 4 672 true del.)					
GASV	N/A	N/A	N/A	0.7	6.9	2.8	(N/A)	N/A	N/A	(0.1)	3.7	1.9
PINDEL	**59.3**	**49.3**	**49.0**	**38.6**	**38.4**	**30.7**	(**9.6**)	**44.2**	**43.4**	(**4.5**)	**44.7**	**41.1**
VariationHunter	0.0	0.0	0.0	2.1	0.1	0.1	(0.0)	0.0	0.0	(0.2)	0.1	0.1
Breakdancer	–	0.0	0.0	2.5	0.0	0.1	(–)	0.0	0.0	(0.0)	0.0	0.0
HYDRA	0.0	0.0	0.0	–	0.0	0.0	(0.0)	0.0	0.0	(–)	0.0	0.0
MoDIL (chr. 1)	3.4	3.1	N/A	0.1	0.1	N/A	(N/A)	N/A	N/A	(N/A)	N/A	N/A
CLEVER	26.4	0.7	0.4	25.9	11.0	4.5	(5.7)	1.3	0.5	(0.9)	4.3	1.9
Length Range 30–49 (3 330 true ins., 3 218 true del.)							(512 true ins., 230 true del.)					
GASV	N/A	N/A	N/A	12.8	19.3	0.6	(N/A)	N/A	N/A	(0.3)	13.0	0.9
PINDEL	**49.5**	31.0	8.5	**43.1**	51.6	5.8	(**4.8**)	23.2	6.9	(**1.3**)	42.2	5.2
VariationHunter	4.3	0.7	0.0	14.2	3.5	0.4	(0.0)	0.2	0.0	(0.4)	2.6	0.0
Breakdancer	–	0.0	0.0	15.0	1.3	0.1	(–)	0.0	0.0	(1.2)	2.2	0.0
HYDRA	0.0	0.0	0.0	–	0.0	0.0	(0.0)	0.0	0.0	(–)	0.0	0.0
MoDIL (chr. 1)	15.1	55.4	N/A	13.7	43.5	N/A	(N/A)	N/A	N/A	(N/A)	N/A	N/A
CLEVER	24.2	**59.5**	**36.9**	33.3	**73.0**	**21.6**	(1.8)	**35.5**	**19.0**	(0.9)	**61.3**	**21.7**
Length Range 50–99 (2 024 true ins., 1 887 true del.)							(256 true ins., 190 true del.)					
GASV	N/A	N/A	N/A	29.6	26.1	0.9	(N/A)	N/A	N/A	(0.5)	27.4	0.5
PINDEL	**41.6**	9.1	1.0	**52.0**	35.3	1.0	(**4.8**)	14.5	1.4	(2.4)	33.7	1.0
VariationHunter	11.5	47.1	1.1	29.3	20.5	1.3	(0.3)	41.0	0.6	(1.1)	18.9	0.0
Breakdancer	34.7	28.4	0.5	43.5	20.1	0.2	(1.9)	3.9	0.0	(**2.8**)	22.1	0.5
HYDRA	0.0	0.0	0.0	–	0.0	0.0	(0.0)	0.0	0.0	(–)	0.0	0.0
MoDIL (chr. 1)	13.9	68.6	N/A	20.6	63.8	N/A	(N/A)	N/A	N/A	(N/A)	N/A	N/A
CLEVER	15.0	**75.0**	**16.3**	51.1	**76.5**	**21.3**	(0.4)	**64.1**	**16.0**	(1.0)	**73.2**	**26.3**
Length Range 100–499 (2 472 true ins., 2 402 true del.)							(164 true ins., 311 true del.)					
GASV				0.8	53.6	1.7				(0.1)	61.7	2.3
PINDEL				**85.8**	40.5	0.2				(**6.8**)	52.7	0.0
VariationHunter				49.2	59.4	2.1				(3.0)	69.8	2.3
Breakdancer				48.6	56.5	0.4				(4.2)	63.7	0.4
HYDRA				85.7	61.3	0.4				(4.9)	67.2	0.4
MoDIL (chr. 1)				45.7	9.1	N/A				(N/A)	N/A	N/A
CLEVER				82.5	**72.4**	**4.7**				(3.5)	**78.1**	**5.2**
Length Range 500–50 000 (629 true ins., 608 true del.)							(1 true ins., 99 true del.)					
GASV				64.6	53.3	0.0				(3.0)	66.7	0.0
PINDEL				66.6	37.3	0.2				(6.2)	57.6	2.0
VariationHunter				71.3	54.8	0.0				(19.0)	62.6	0.0
Breakdancer				**81.5**	52.3	0.0				(5.1)	65.7	0.0
HYDRA				61.3	62.2	3.5				(**29.5**)	71.7	0.0
MoDIL (chr. 1)				–	0.0	N/A				(N/A)	N/A	N/A
CLEVER				79.1	**64.5**	**5.8**				(3.0)	**77.8**	**6.1**

bp) and larger ($\geq$ 100 bp) indels. Most importantly, our approach detects the greatest amounts of variants which are missed by all other approaches. Nearly 25% of all medium-sized indels can only be discovered by our tool while none of the other approaches makes more than 6% of such unique predictions.

Our approach builds on two key elements: first, an algorithm that enumerates maximal, statistically contradiction-free ensembles as max-cliques in read alignment graphs in short time and, second, a sound statistical procedure that reliably calls max-cliques which indicate variants. Our approach is generic with respect to choices of variants; max cliques in the read alignment graphs can also reflect other variants such as inversions or translocations. As future work, we are planning to predict inversions and to incorporate split read information in a unifying approach.

References

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

Albers, C. A., Lunter, G., MacArthur, D. G., McVean, G., Ouwehand, W. H., and Durbin, R. (2011). Dindel: accurate indel calls from short-read data. *Genome Research*, **21**(6), 961–973. PMID: 20980555.

Alkan, C., Kidd, J. M., Marques-Bonet, T., Aksay, G., Antonacci, F., *et al.* (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature Genetics*, **41**(10), 1061–1067. PMID: 19718026.

Alkan, C., Coe, B. P., and Eichler, E. E. (2011). Genome structural variation discovery and genotyping. *Nat Rev Genet*, **12**(5), 363–376.

AppliedBiosystems (2009). The SOLID system: Next-generation sequencing. www.appliedbiosystems.com.

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., *et al.* (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, **456**(7218), 53–59.

Bravo, H. C. and Irizarry, R. A. (2010). Model-based quality assessment and base-calling for second-generation sequencing data. *Biometrics*, **66**(3), 665–674. PMID: 19912177.

Campbell, P. J., Stephens, P. J., Pleasance, E. D., O'Meara, S., Li, H., *et al.* (2008). Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nat Genet*, **40**(6), 722–729.

Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., *et al.* (2009a). Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Methods*, **6**(9), 677–681.

Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., *et al.* (2009b). BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Meth*, **6**(9), 677–681.

Chiang, D. Y., Getz, G., Jaffe, D. B., O'Kelly, M. J. T., Zhao, X., *et al.* (2009). High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat Methods*, **6**(1), 99–103.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., *et al.* (2009). Real-Time DNA sequencing from single polymerase molecules. *Science*, **323**(5910), 133 –138.

Fishburn, P. C. (1985). *Interval orders and interval graphs: a study of partially ordered sets*. John Wiley & Sons.

Hach, F., Hormozdiari, F., Alkan, C., Hormozdiari, F., Birol, I., *et al.* (2010). mrsfast: a cache-oblivious algorithm for short-read mapping. *Nat Methods*, **7**(8), 576–577.

Hormozdiari, F., Alkan, C., Eichler, E. E., and Sahinalp, S. C. (2009). Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Research*, **19**(7), 1270–1278. PMID: 19447966.

Hormozdiari, F., Hajirasouliha, I., Dao, P., Hach, F., Yorukoglu, D., Alkan, C., Eichler, E. E., and Sahina lp, S. C. (2010). Next-generation VariationHunter: combinatorial algorithms for tran sposon insertion discovery. *Bioinformatics*, **26**(12), i350–i357.

Itsara, A., Cooper, G. M., Baker, C., Girirajan, S., Li, J., *et al.* (2009). Population analysis of large copy number variants and hotspots of human genetic disease. *Am J Hum Genet*, **84**(2), 148–161.

Korbel, J. O., Abyzov, A., Mu, X. J., Carriero, N., Cayting, P., *et al.* (2009). PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biology*, **10**(2), R23. PMID: 19236709.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, **10**(3), R25.

Lee, S., Hormozdiari, F., Alkan, C., and Brudno, M. (2009). MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat Meth*, **6**(7), 473–474.

Levy, S., Sutton, G., Ng, P. C., Feuk, L., Halpern, A. L., *et al.* (2007). The diploid genome sequence of an individual human. *PLoS Biol*, **5**(10), e254.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, **25**(14), 1754–1760. PMID: 19451168.

Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*, **11**(5), 473 –483.

Li, H., Ruan, J., and Durbin, R. (2008). Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Res*, **18**(11), 1851–1858.

Medvedev, P., Stanciu, M., and Brudno, M. (2009). Computational methods for discovering structural variation with next-generation sequencing. *Nat Meth*, **6**(11s), S13–S20.

Mills, R., Pittard, W., Mullaney, J., Farooq, U., Creasy, T., *et al.* (2011). Natural genetic variation caused by small insertions and deletions in the human genome. *Genome Research*, **21**, 830–839.

Mills, R. E., Luttig, C. T., Larkins, C. E., Beauchamp, A., Tsui, C., *et al.* (2006). An initial map of insertion and deletion (indel) variation in the human genome. *Genome Res*, **16**(9), 1182–1190.

Quinlan, A. R., Clark, R. A., Sokolova, S., Leibowitz, M. L., Zhang, Y., *et al.* (2010). Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Research*, **20**(5), 623 –635.

Sindi, S., Helman, E., Bashir, A., and Raphael, B. J. (2009). A geometric approach for classification and comparison of structural variants. *Bioinformatics*, **25**(12), i222–i230. PMID: 19477992 PMCID: 2687962.

Sudmant, P. H., Kitzman, J. O., Antonacci, F., Alkan, C., Malig, M., *et al.* (2010). Diversity of human copy number variation and multicopy genes. *Science*, **330**(6004), 641–646.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061–1073.

The International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, **437**(7063), 1299–1320.

Wasserman, L. (2004). *All of Statistics*. Springer.

Ye, K., Schulz, M. H., Long, Q., Apweiler, R., and Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**(21), 2865–2871.

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