BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 24 2013, pages 3143–3150
doi:10.1093/bioinformatics/btt556

Sequence analysis

Advance Access publication September 25, 2013

MATE-CLEVER: Mendelian-inheritance-aware discovery and genotyping of midsize and long indels

Tobias Marschall[1,*], Iman Hajirasouliha[2] and Alexander Schönhuth[1,*]

[1]Centrum Wiskunde & Informatica (CWI), Life Sciences Group, Science Park 123, Amsterdam 1098 XG,
The Netherlands and Department of Computer Science and [2]Brown University, Providence, Rhode Island 02906, USA

Associate Editor: Michael Brudno

ABSTRACT

Motivation: Accurately predicting and genotyping indels longer than 30 bp has remained a central challenge in next-generation sequencing (NGS) studies. While indels of up to 30 bp are reliably processed by standard read aligners and the Genome Analysis Toolkit (GATK), longer indels have still resisted proper treatment. Also, discovering and genotyping longer indels has become particularly relevant owing to the increasing attention in globally concerted projects.

Results: We present MATE-CLEVER (Mendelian-inheritance-AtTEntive CLique-Enumerating Variant findER) as an approach that accurately discovers and genotypes indels longer than 30 bp from contemporary NGS reads with a special focus on family data. For enhanced quality of indel calls in family trios or quartets, MATE-CLEVER integrates statistics that reflect the laws of Mendelian inheritance. MATE-CLEVER's performance rates for indels longer than 30 bp are on a par with those of the GATK for indels shorter than 30 bp, achieving up to 90% precision overall, with >80% of calls correctly typed. In predicting *de novo* indels longer than 30 bp in family contexts, MATE-CLEVER even raises the standards of the GATK. MATE-CLEVER achieves precision and recall of ~63% on indels of 30 bp and longer versus 55% in both categories for the GATK on indels of 10–29 bp. A special version of MATE-CLEVER has contributed to indel discovery, in particular for indels of 30–100 bp, the 'NGS twilight zone of indels', in the Genome of the Netherlands Project.

Availability and implementation: http://clever-sv.googlecode.com/

Contact: tm@cwi.nl or as@cwi.nl

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on March 17, 2013; revised on August 26, 2013; accepted on September 20, 2013

1 INTRODUCTION

More than 6 years after its introduction, *next-generation sequencing* (*NGS*) has become standard technology. Read length is steadily increasing and so is sequencing speed, at an overall still decreasing sequencing cost. One of the most evident advantages of NGS over array-based approaches is that it has enabled studying genetic variation beyond single nucleotide polymorphisms (SNPs) at both a larger scale and finer resolution. Several large-scale projects addressing this are under way [e.g. The 1000 Genomes Project Consortium (2010), The Genome of the Netherlands (GoNL) Project Consortium], which have accumulated tera-scale amounts of NGS data. The goal is to discover and genotype variants in thousands of individuals at single base-pair resolution, and, in a second step, to classify them according to population structure and phenotype, such as susceptibilities for diseases.

For simplicity, we will refer to all insertion and deletion variants as *indel variants*, or simply *indels*—we are aware of the occasional clash with nomenclature for structural variants. We will refer to *indel discovery* as prediction of indels without predicting the zygosity status of indel alleles, and to *genotyping indels* as determining (e.g. computationally predicting) this zygosity status.

Motivation. As one particular example of a nationwide directed effort, the GoNL Project puts particular emphasis on drawing links between population structure and inheritability. In the frame of this project, 769 Dutch individuals, consisting of 231 mother–father–child trios, 11 monozygotic- and 8 dizygotic-twin quartets, have been sequenced. In the analysis, indels play a major role—the obvious reason is that only NGS has made large-scale discovery of indels truly possible.

Also for other large-scale projects (The 1000 Genomes Project Consortium, 2010; The International Cancer Genome Consortium, 2010), mapping and categorizing indels is of utmost relevance. However, both indel discovery and genotyping technology still lag considerably behind the advances made in sequencing technology itself (Alkan *et al.*, 2011). Existing indel discovery tools still leave much room for improvement. On top of that, with only very few exceptions, they do not offer genotyping as an option. An analysis of existing state-of-the-art trio variant callsets (e.g. that of the 'platinum genome trio' issued by Illumina, see Appendix A.2 in the Supplementary Material for more details) points out that a surprisingly large fraction of indel calls violate the Mendelian laws. This decisively differs from SNPs, which have been genotyped soundly and reliably.

In fact, genotyping SNPs has been standard for many years. Since 2011, smaller indels can also be reliably handled. The *Unified Genotyper (UG)*, a tool from the *Genome Analysis ToolKit (GATK)* (DePristo *et al.*, 2011; McKenna *et al.*, 2010), has been providing the corresponding technology. We will demonstrate the GATK-UG genotypes between 80 and 90% of indels correctly in the Results section. However, its performance sharply drops for indels of 30 bp and longer. The reason is that one commonly runs the GATK on standard read alignments, e.g. as delivered by BWA (Li and Durbin, 2009), Bowtie2 (Langmead and Salzberg, 2012) or Stampy (Lunter and Goodson, 2011).

*To whom correspondence should be addressed.

For indels longer than 30 bp, principled, statistically and algorithmically sound approaches have not been described.

Our Contribution. We present a novel approach, *MATE-CLEVER (Mendelian-inheritance-AtTEntive CLique-Enumerating Variant findER)* for sound discovery and genotyping of indels longer than 30 bp from NGS reads that were generated through contemporary library protocols [For old or non-standard protocols, we claim that we can discover and genotype indels longer than two times the standard deviation (stddev) of the fragment length distribution, which amounts to 30 on current protocols.]. MATE-CLEVER provides a novel (Bayesian) probabilistic framework to compute the probabilities that an indel allele is *homozygous*, *heterozygous* or *not present*. In case of a family, it integrates prior probabilities that reflect the laws of Mendelian inheritance, which yields enhanced performance rates when combinedly genotyping family members. Overall, MATE-CLEVER achieves performance rates on longer indels that are as favorable as those of the GATK-UG on small indels. Thus, our approach considerably raises the limits of sound indel genotyping.

For accomplishing this, we have combined the two most recent approaches of ours, CLEVER (Marschall *et al.*, 2012) and LASER (Marschall and Schönhuth, 2013), into a hybrid approach that specifically targets indels longer than 30 bp. While CLEVER has proven to be able to discover indels of 30–100 bp—sometimes referred to as the (*NGS*) *twilight zone of indels*—at highly favorable recall and precision rates, LASER allows us to re-evaluate calls and adds split-read alignment information, which leads to enhanced genotyping and high breakpoint resolution.

Related Work. Structural variant (SV) discovery tools can be divided into four large classes of approaches, for each of which we restrict ourselves to naming a few state-of-the-art tools.

1. *Internal segment size (also: insert size)-based approaches* identify groups of paired-end reads whose alignments exhibit abnormal internal segment lengths with respect to a background distribution. That is, they consider the distance between the two alignments of a read pair (called internal segment size or insert size). Groups of alignment pairs with deviating distance suggest the presence of indel breakpoints in the internal segment. We will refer to such alignment pairs as *spanning alignments* (see Fig. 1, top). Examples of methods using this type of signal are Breakdancer (Chen *et al.*, 2009), VariationHunter (Hormozdiari *et al.*, 2009), HYDRA (Quinlan *et al.*, 2010), PEMER (Korbel *et al.*, 2009), MoDIL (Lee *et al.*, 2009) as well as our tool CLEVER (Marschall *et al.*, 2012). It is characteristic for these approaches to successfully predict indels longer than 100 bp. However, methods that impose a hard threshold and only work on 'discordant' reads, like BreakdancerMax, VariationHunter, HYDRA and PEMER cannot detect smaller indels. [Note that the initial Breakdancer release consisted of BreakdancerMini and BreakdancerMax. Maintenance/development of BreakdancerMini has been stopped, and current releases only contain BreakdancerMax.] MoDIL and BreakdancerMini can, in principle, detect shorter indels, but the former is prohibitively slow (see discussion in Marschall *et al.*, 2012), and the implementation of the latter is no longer maintained. CLEVER was designed to process *all* alignments in a short time and thus achieves good performance also for indels in the 'twilight zone' (30–100 bp). For insert size-based approaches, placement of breakpoints is commonly rather little accurate.

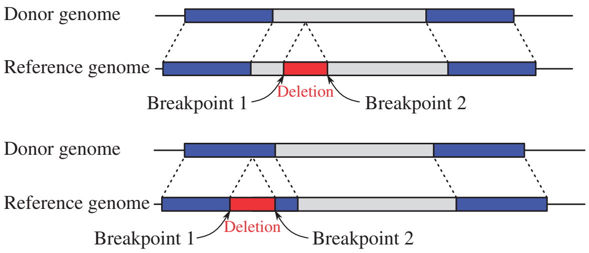

Fig. 1. Blue rectangles represent sequenced ends of a fragment, whereas gray rectangles represent the (unsequenced) internal segment. In the two subfigures (top and bottom), parts of the reference genome (marked in red) are deleted from the donor genome. Top: *Spanning alignment pair:* The breakpoints of the deletion are located in the internal segment of the paired-end read alignment (i.e. the gray part). Bottom: *Breakpoint-covering alignment:* The breakpoints are located within the alignment of one read end (i.e. the blue part), which yields a split-read alignment

2. *Split-read aligners* aim at aligning reads across the breakpoints of insertions and deletions. When the alignment of a read end contains an indel breakpoint, we refer to it as *breakpoint-covering alignment* (see Fig. 1, bottom). Examples for split-read methods are PINDEL (Ye *et al.*, 2009), SplazerS (Emde *et al.*, 2012) as well as our tool LASER (Marschall and Schönhuth, 2013). Split-read aligners predict breakpoints at single-base-pair resolution. However, most of them apply predominantly for indels up to 30 bp. Longer indels can be challenging for split-read aligners if the alignment of split parts of reads is not properly guided.

3. *Coverage-based* approaches aim at detecting deletions and duplications by measuring amounts of reads mapped to locations. Examples are CNVer (Medvedev *et al.*, 2010) and CNVnator (Abyzov *et al.*, 2011). Although coverage-based approaches are the only reliable technique to predict large duplications, they only work for very large deletions and duplications.

4. *De novo assembly* methods focus on reconstructing sequences without using a reference genome. A few more well-known examples are ALLPATHS (Gnerre *et al.*, 2011), SOAPDenovo (Li *et al.*, 2010) and VELVET (Zerbino and Birney, 2008). An approach that focuses exclusively on novel sequence insertions is NovelSeq (Hajirasouliha *et al.*, 2010).

Our method, MATE-CLEVER, falls among the so-called *hybrid methods*, as it makes use of both *insert size* signal and *split-read* information. Such hybrid methods have established a new class of approaches and have arisen in the literature only from 2012 onward. Examples are DELLY (Rausch *et al.*, 2012), SVSeq2 (Zhang *et al.*, 2012) and PRISM (Jiang *et al.*, 2012).

The only discovery method for larger indels currently available that addresses *family settings* is CommonLAW (Hormozdiari *et al.*, 2011), which draws from VariationHunter (Hormozdiari *et al.*, 2009) as a core approach. CommonLAW collects information on frequencies of variants that are supported by combinations of family members in a preprocessing step. It uses this information in the form of *prior weights* in a combinatorial algorithm to predict variants parsimoniously.

Although approaches that aim at genotyping indels have rarely been described, there have been reliable techniques for shorter indels since 2011. Examples for processing indels of length up to 30 bp are the above-mentioned UG (DePristo *et al.*, 2011) from the GATK (McKenna *et al.*, 2010) and the earlier approach DINDEL (Albers *et al.*, 2011), whose core ideas initially inspired the GATK-UG. An option for indels from 125 bp or longer is GASV-Pro (Sindi *et al.*, 2012), which reportedly achieves sufficiently reliable classification rates on very long indels.

Reliable pipelines for genotyping indels that are ~30–100 bp long, in particular in the context of family settings, have not been described in the literature so far.

2 METHODS

2.1 Laws of Mendelian inheritance

According to Mendel's laws, a child inherits exactly one allele from each parent. Egg and sperm cells each contain one set of chromosomes, representing a *recombination* of the two sets of chromosomes present in each parent. During recombination, it is determined which alleles are passed on to the child. Let 0,1,2 be placeholders for 'variant *not present*', '*heterozygous* variant' and '*homozygous* variant', respectively. We write X_{gh} with $g, h \in \{0, 1, 2\}$ and $g \leq h$ to refer to variants that are 'g' in one parent and 'h' in the other one. For example, X_{01} refers to variants that are heterozygous in either the mother or father, but not present in the other parent. Analogously, we write Y_j with $j \in \{0, 1, 2\}$ to denote the event that the child has genotype *j*. Assuming that each allele is equally likely to be transmitted to the child, the Mendelian laws can be cast statistically as

$$\begin{array}{lll} \mathbf{P}(Y_0|X_{01}) = \frac{1}{2} & \mathbf{P}(Y_1|X_{01}) = \frac{1}{2} & \mathbf{P}(Y_2|X_{01}) = 0 \\ \mathbf{P}(Y_0|X_{11}) = \frac{1}{4} & \mathbf{P}(Y_1|X_{11}) = \frac{1}{2} & \mathbf{P}(Y_2|X_{11}) = \frac{1}{4} \\ \mathbf{P}(Y_0|X_{02}) = 0 & \mathbf{P}(Y_1|X_{02}) = 1 & \mathbf{P}(Y_2|X_{02}) = 0 \\ \mathbf{P}(Y_0|X_{12}) = 0 & \mathbf{P}(Y_1|X_{12}) = \frac{1}{2} & \mathbf{P}(Y_2|X_{12}) = \frac{1}{2} \\ \mathbf{P}(Y_0|X_{22}) = 0 & \mathbf{P}(Y_1|X_{22}) = 0 & \mathbf{P}(Y_2|X_{22}) = 1 \end{array} \tag{1}$$

When discovering genetic variations in a trio rather than a single individual, one can take advantage of Mendel's laws. A variant found in one of the parents has (at least) a 50% probability of being present in the child. This translates to the *conditional* probability of a variant being present in the child, given its presence in one of the parents, being higher than the *a priori* probability, which is not conditioned on any prior knowledge.

2.2 MATE-CLEVER workflow

Step 1: Running CLEVER. We run our tool CLEVER (Marschall *et al.*, 2012) on each individual independently. CLEVER processes *all* alignments, including also concordant ones, which allows it to also detect indels shorter than 100 bp. The output of this step is a set of deletions $\mathcal{D}_{\text{clever}}$.

Step 2: Defining Regions of Interest. At this stage, deletions still may have remained undiscovered in single individuals. So, for each $d \in \mathcal{D}_{\text{clever}}$, we declare a window of ±1000 bp around *d*, a *region of interest* in *every individual*, independently of in which individual *d* was originally discovered.

Step 3: Extracting Reads of Interest. We extract all paired-end reads where one end aligns in a region of interest. Owing to pooling deletions from all family members when determining these regions, this may also include paired-end reads from individuals who do not have a CLEVER deletion in the region of interest themselves. The goal of this is to discover breakpoint-covering reads, where one end has to be *split-aligned* and therefore has remained unaligned by the (standard) read aligner in use (Li and Durbin, 2009). Let $\Sigma = \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{C}\}$ be the DNA alphabet and let ℓ be the read length. That is, each paired-end read can be considered a pair of sequences of length ℓ, i.e. an element of $\Sigma^\ell \times \Sigma^\ell$. Correspondingly, we write $\mathcal{R} \subset \Sigma^\ell \times \Sigma^\ell$ for the set of all extracted read pairs, from all individuals.

Step 4: Generating (Split) Alignments with LASER. We *align* all reads $r \in \mathcal{R}$ using our *read aligner* LASER (Marschall and Schönhuth, 2013). LASER determines *both normal alignments*, containing only short indels *and split alignments*, indicating long indels. We report up to 50 alignments per read end. In case of ambiguous indel placements, we only report the leftmost one. We write $\mathcal{A}_1(r)$ and $\mathcal{A}_2(r)$ to denote the set of alignments of the first and second read end of $r \in \mathcal{R}$. We define $\mathcal{A}(r) = \mathcal{A}_1(r) \cup \mathcal{A}_2(r)$ and $\mathcal{A}(\mathcal{R}) = \bigcup_{r\in\mathcal{R}} \mathcal{A}(r)$. For each alignment $A \in \mathcal{A}(r)$, LASER estimates the probability of it indicating the correct placement of *r*, written $\mathbf{P}(A)$, based on phred scores and empirical indel statistics; see Appendix D for details.

Step 5: Refining the List of Putative Deletions. We now refine the set of putative deletions $\mathcal{D}_{\text{clever}}$ to create a list of candidate deletions $\mathcal{D}_{\text{cand}}$ that are supported by *both spanning and breakpoint-covering alignments*, i.e. are supported by both CLEVER and split alignments from LASER. Here, we again aim at a high sensitivity: we prefer to err on the side of including too many rather than too few candidates. We will purge bad candidates in a later step. Let $\mathcal{D}_{\text{split}}$ be the set of all deletions supported by split alignments generated in Step 4. We compute the *expected support* $S_\mathcal{R}(d)$ of a deletion $d \in \mathcal{D}_{\text{split}}$ as follows:

$$S_\mathcal{R}(d) := \sum_{A\in\mathcal{A}(\mathcal{R})} \mathbb{I}_d(A) \cdot \mathbf{P}(A)$$

where $\mathbb{I}_d(A)$ is an indicator that is 1 when alignment *A* contains deletion *d* and 0 otherwise. To be rather permissive, we retain all deletions with an expected support of 0.5 and above and thus set

$$\mathcal{D}'_{\text{split}} := \{d \in \mathcal{D}_{\text{split}} : S_\mathcal{R}(d) \geq 0.5\}$$

Next, we filter out deletions that are not similar to any deletion from $\mathcal{D}_{\text{clever}}$. Formally, we define the set of candidate deletions as follows:

$$\mathcal{D}_{\text{cand}} := \{d \in \mathcal{D}'_{\text{split}} : \text{there exists } d' \in \mathcal{D}_{\text{clever}} \text{ such that } \Delta_L(d, d') < T_L \text{ and } \Delta_O(d, d') < T_O\} \tag{2}$$

where $\Delta_L(d, d')$ is the length difference between *d* and *d'* and $\Delta_O(d, d')$ is the offset, i.e. the distance of their center points. CLEVER predictions are based on internal segment statistics rather than on alignments. Therefore, position and length can differ from the true deletion. The thresholds T_L and T_O have to allow for enough flexibility to take that into account without creating too many spurious hits. We found that setting $T_L = 20$, just slightly above the insert size distribution's stddev, and $T_O = 100$, to a distance that is unlikely to produce random hits, works well in practice.

Step 6: Recalibrating Alignment Scores. Because of empirical indel statistics, the probability $\mathbf{P}(A)$ tends to be small if *A* is an alignment that supports a deletion (either through too long insert size, in case of a spanning alignment, or through a split, in case of a breakpoint-covering alignment). Because we are ready to believe in deletions $d \in \mathcal{D}_{\text{cand}}$, we increase the probabilities $\mathbf{P}(A)$ for alignments supporting deletions $d \in \mathcal{D}_{\text{cand}}$ as follows.

All deletions $d \in \mathcal{D}_{\text{cand}}$ now incur only the minimum phred-scaled cost of 1, whereas all other deletions retain their original costs. To be more precise, a deletion $d' \notin \mathcal{D}_{\text{cand}}$ retains a cost of $C_{\text{del}}(L(d'))$ as defined in Appendix D. Using the updated deletion costs, we re-compute the posterior distribution over all alternative alignments (as described in

Appendix D). This systematically increases $\mathbf{P}(A)$ for alignments A indicating deletions $d \in \mathcal{D}_{\mathrm{cand}}$, but not for A indicating deletions d not in $\mathcal{D}_{\mathrm{cand}}$. From a Bayesian point of view, this procedure corresponds to updating our prior belief in alignments A into posterior probabilities by incorporation of the additional evidence provided by $\mathcal{D}_{\mathrm{cand}}$. We then compute the *most likely alignment pair*:

$$(A_1^*(r), A_2^*(r)) := \underset{\substack{(A_1, A_2) \\ \in \mathcal{A}_1(r) \times \mathcal{A}_2(r)}}{\arg\max}\ \mathbf{P}(A_1) \cdot \mathbf{P}(|I(A_1, A_2)|) \cdot \mathbf{P}(A_2)$$

where $\mathbf{P}(|I(A_1, A_2)|)$ is the empirical probability (determined from uniquely mappable reads) to observe an internal segment of size $I(A_1, A_2)$. We discard all other alignments. We also discard alignments where $\mathbf{P}(A) < 1 - 0.001$, equivalent to a phred-scaled mapping quality of at least 30.

Step 7: Genotyping each Individual. For each deletion, we predict the genotype of each individual based on a prior belief p^{prior}, evidence from covering alignments, and evidence from spanning alignment pairs. The prior belief is a global user-specified input parameter. It can be used to adjust the trade-off between precision and recall. We process all $d \in \mathcal{D}_{\mathrm{cand}}$ ordered (decreasingly) by expected support $S_{\mathcal{R}}(d)$. In the following, let $C(d)$ be the arithmetic mean of start and end position of d.

Spanning alignment pairs: We assume insert sizes to be normally distributed with mean μ and stddev σ [see Marschall *et al.* (2012) for further discussion of this assumption]. A read r is said to *support* the deletion d if

$$d \subset I(A_1^*(r), A_2^*(r)) \quad \text{and} \quad |I(A_1^*(r), A_2^*(r))| - \mu > |d|/2,$$

where $I(A_1^*(r), A_2^*(r))$ is the set of positions that constitute the internal segment between the two alignments. That is, deletion d lies in the internal segment and the internal segment length is closer to $\mu + |d|$ than to μ. Owing to the symmetry of the null distribution $\mathcal{N}_{\mu,\sigma}$, the latter is equivalent to that the deletion is more likely to be present than absent relative to r. Read *r contradicts* the deletion d if

$$C(d) \in I(A_1^*(r), A_2^*(r)) \quad \text{and} \quad |I(A_1^*(r), A_2^*(r))| - \mu \leq |d|/2.$$

That is, the center point of d lies in the internal segment, whereas in this case d is more likely to be absent than present relative to r. Note that for contradicting reads d itself does not necessarily lie in the internal segment, which is an obvious requirement. Because a deletion reduces to a single breakpoint in the donor genome, considering only center points for contradicting reads is analogous to considering the entire deletion for supporting reads. For accurate genotyping, we have to take ‘crosstalk’ between supporting and contradicting reads into account. Therefore, we determine $p_{\mathrm{FP}}^{\mathrm{insert}} := \int_{\mu+|d|}^{\infty} \mathcal{N}_{\mu,\sigma}(x)dx$ as the probability of a *false positive* (a contradicting read being misclassified as supporting) and $p_{\mathrm{FN}}^{\mathrm{insert}} = p_{\mathrm{FP}}^{\mathrm{insert}}$ as the probability of a *false negative* (a supporting read misclassified as contradicting).

Let n_S and n_C be the number of supporting and contradicting reads, respectively, and let $n := n_S + n_C$. Let $\mathcal{B}_{(n,p)}(k)$ be the probability that out of n samples, each of which, with probability p, has a special label, k samples have the label. This reflects a common binomial distribution. We then determine (0 for indel not present, 1 for heterozygous and 2 for homozygous)

$$\begin{aligned} p_0^{\mathrm{ins}} &\propto \mathcal{B}_{n,p}(n_S),\ p = p_{\mathrm{FP}}^{\mathrm{insert}} \\ p_1^{\mathrm{ins}} &\propto \mathcal{B}_{n,p}(n_S),\ p = 0.5p_{\mathrm{FP}}^{\mathrm{insert}} + 0.5(1 - p_{\mathrm{FN}}^{\mathrm{insert}}) \qquad (3) \\ p_2^{\mathrm{ins}} &\propto \mathcal{B}_{n,p}(n_S),\ p = 1 - p_{\mathrm{FN}}^{\mathrm{insert}} \end{aligned}$$

as insert-size based probabilities for a deletion being not present, heterozygous and homozygous.

Breakpoint-covering alignments: We determine the number of read ends supporting the deletion

$$n_s = \sum_{r \in \mathcal{R}} \mathbb{1}_d(A_1^*(r)) + \mathbb{1}_d(A_2^*(r))$$

where $\mathbb{1}_d(A)$ is 1 if alignment A contains the deletion d and 0 otherwise. We also count the number n of all reads r whose alignments overlap $C(d)$ (hence, $n_C : n - n_S$ is the number of alignments that contradict d). We set $p_{\mathrm{FN}}^{\mathrm{split}} = p_{\mathrm{FP}}^{\mathrm{split}} = 0.01$. This is rather conservative—recall that the alignments were filtered at a 0.001 level in Step 6. We proceed analogously as for spanning alignments (see Equation 3), but now use n and n_S, as obtained here. This yields probabilities $p_0^{\mathrm{split}}, p_1^{\mathrm{split}}, p_2^{\mathrm{split}}$.

We also consider $p_0^{\mathrm{prior}} := 1 - p_{\mathrm{prior}}$, $p_1^{\mathrm{prior}} = p_2^{\mathrm{prior}} := p_{\mathrm{prior}}/2$, which expresses our prior belief in the zygosity status of deletions. To achieve a high precision, we set $p_{\mathrm{prior}} = 0.001$.

Let $p_i := p_i^{\mathrm{prior}} \cdot p_i^{\mathrm{ins}} \cdot p_i^{\mathrm{split}}$ for $i = 0, 1, 2$. We finally determine the type as

$$\underset{0,1,2}{\arg\max}\, \{p_0, p_1, p_2\}$$

which corresponds to the genotype with the highest posterior probability.

Recall that we process deletions d ordered by expected support $S_{\mathcal{R}}(d)$. When processing subsequent deletions, all alignments already in use for genotyping a higher-ranked deletion d will be ignored.

Step 8: Finding De Novo Deletions. *De Novo* deletions, i.e. deletions present only in the child, but not in the parents, are rare but usually of utmost interest. Their detection is difficult for two reasons. On one hand, they are absent in the parents and heterozygous in the child, i.e. only one out of six alleles represents them. This results in considerably less power for detecting them. On the other hand, inherited deletions that are heterozygous in only one parent are prone to be mistaken as *de novo*, if local coverage in the respective parent is low.

To avoid such spurious calls, we further filter all deletions that were genotyped as present only in the child, by not only requiring strong evidence for its presence in the child, *but also* strong evidence that it is absent in the parents. Let $T_{\text{de-novo}} = 10^{-5}$; we recall that p_0 is the probability (from Step 7) that a deletion is not present in an individual. All deletions with $p_0 < T_{\mathrm{de-novo}}$ in the child *and* with $1 - p_0 < T_{\mathrm{de-novo}}$ in both parents are reported as *de novo*. All other deletions are passed on to the next processing step.

Step 9: Family-Structure-Aware Genotyping. Finally, we combine individual probability distributions (p_0, p_1, p_2) (0 for not present, 1 for heterozygous and 2 for homozygous) from all family members into a combined probability distribution $(p_{ghj}; g, h, j \in \{0, 1, 2\})$. For example, $p_{021} = p_0^{mo} \cdot p_2^{fa} \cdot p_1^{ch}$, where we index with ${}^{mo}, {}^{fa}, {}^{ch}$ for the different family members, is the probability that the deletion is not present in the mother, homozygous in the father and heterozygous in the child. We combine these probabilities with a Mendelian prior $q_{ghj}; g, h, j \in \{0, 1, 2\}$ where

$$q_{ghj} \propto \begin{cases} \mathbf{P}(Y_j | X_{gh}) & ghj \text{ respects the Mendelian laws} \\ 0 & \text{otherwise} \end{cases}$$

See (1) for $\mathbf{P}(Y_j | X_{gh})$. We report

$$\underset{ghj}{\arg\max}\, \{q_{ghj} \cdot p_{ghj}\}$$

as the final Mendelian-law-corrected genotype.

3 EVALUATION

3.1 ‘Venter’s Family’—a Trio Benchmark

We derive Mendelian-inheritance-compliant annotations for a virtual family from the full set of Craig Venter’s variants (Levy *et al.*, 2007). We opt for proceeding this way for three reasons. (i) For indels, there are no trio benchmark datasets available (see also the discussion in Appendix A.2). (ii) The Venter variants make an encompassing amount of real annotations.

In particular, we only have to simulate reads, but not variants [see also Marschall *et al.* (2012) for more arguments on good benchmark datasets]. (3) Dividing these variants into three (overlapping) subsets, one each for a mother, a father and the child, does not lead to a significant reduction of quantities of variants in one of the single individuals because Mendel's laws imply that the vast majority of annotations are shared by at least two individuals.

Formally, let Ω be the entire set of annotations. We divide (see section 'Laws of Mendelian inheritance' for notation)

$$\Omega = X_{01} \dot{\cup} X_{11} \dot{\cup} X_{02} \dot{\cup} X_{12} \dot{\cup} X_{22} \tag{4}$$

We also set aside a certain amount of *de novo variants* X_{00}. We assume that fractions of variants of equal zygosity are the same in both mother and father. Similarly, we divide $\Omega = Y_0 \dot{\cup} Y_1 \dot{\cup} Y_2$ into subsets.

The (computational) problem is to partition a given set of annotations Ω into subsets

$$(Y_0 \cap X_{10}) \dot{\cup} (Y_0 \cap X_{11}) \dot{\cup} \ldots \dot{\cup} (Y_2 \cap X_{22}) \tag{5}$$

without violating *Mendel's laws* and such that the resulting callset makes a reasonable SV discovery and genotyping benchmark. Refer to Appendix A.1 for how we solve this problem geometrically. Owing to the large number of heterozygous deletions and because heterozygous deletions are harder to discover than homozygous deletions, the resulting benchmark is relatively difficult. We included (unrealistically) large amount of *de novo deletions* in the benchmark because we need a statistically sufficient mass of *de novo* annotations for benchmarking properly.

Using the resulting trio benchmark annotations, which include all SNPs, indels, mixed variants and inversions, i.e. all variants one can download from Levy *et al.* (2007), we simulated reads using the read simulator SimSeq (Earl *et al.*, 2011). Mean fragment size was set to $\mu = 500$ and stddev to $\sigma = 15$, which reflects common standards and was chosen as it resembles the data in the GoNL project (Boomsma *et al.*, 2013). We generated two different datasets of reads, one of which amounts to 12× coverage (which meets the GoNL Project standards) and the other one has 30× coverage. We then aligned all reads with BWA (Li and Durbin, 2009), with default parameters in the 'aln' step and parameters '`-n25 -N25`' in the 'sampe' step to allow for up to 25 alternative alignments per read.

3.2 Results: Venter simulated reads

For comparing MATE-CLEVER with state-of-the-art approaches, we selected the following three tools: (i) *Unified Genotyper* (abbreviated UG or simply GATK), which sets the *de facto* standard for both population- and multi-sample-aware genotyping (DePristo *et al.*, 2011); (ii) *PINDEL* as a most prominent, state-of-the-art split-read approach that sets the *de facto* standard in split-aligning reads (Ye *et al.*, 2009); and (iii) *CommonLAW* as the only indel discovery tool available, apart from the GATK, that is distinctly trio-aware (Hormozdiari *et al.*, 2011).

We have furthermore evaluated DELLY (Rausch *et al.*, 2012) and PRISM (Jiang *et al.*, 2012), both of which, like MATE-CLEVER, are hybrid approaches. However, DELLY does not address discovery of indels in the length range considered here (Tobias Rausch, personal communication). Moreover, both tools address neither genotyping of indels nor the integration of family information during discovery. See Appendix B for corresponding results. We furthermore refer to the CLEVER article (Marschall *et al.*, 2012) for performance statistics for Breakdancer (Chen *et al.*, 2009), GASV (Sindi *et al.*, 2009), HYDRA (Quinlan *et al.*, 2010), VariationHunter (Hormozdiari *et al.*, 2009), SVSeq2 (Zhang *et al.*, 2012) and MoDIL (Lee *et al.*, 2009).

Because CommonLAW is internal segment size-based, we evaluate its calls with relaxed distance thresholds and compare its results with MATE-CLEVER calls evaluated in the same way in a separate table to avoid confusion and/or unfair comparisons. The comparison of MATE-CLEVER, the GATK and PINDEL (with strict criteria) is shown in Table 1, whereas the comparison of MATE-CLEVER and CommonLAW (with relaxed criteria) is displayed in Table 2; the precise criteria and an explanation of evaluation metrics are provided in the next section. A detailed overview of MATE-CLEVER's genotyping performance is given in Figure 2.

Table 3 shows the performance of MATE-CLEVER, UG and PINDEL in terms of making *de novo* predictions. *De novo* deletions only exist in the child, hence cannot be inherited, but must have come into existence during mating. *De novo* variants, including SNPs, are rare, but are also of great interest, because they can help to explain the mechanisms behind creation of new genetic variation. We recall that we included amounts of *de novo* calls in our benchmark that overestimate true amounts because we need a statistically sufficient mass for evaluation.

3.3 Evaluation metrics

In Table 1, we count a prediction as true positive if it matches a true annotation at most 20-bp distance, with at most 10 bp difference in length. For a relaxed evaluation of CommonLAW (Table 2), a true positive is a prediction that matches a true annotation at a distance of at most 100-bp, with at most 100 bp difference in length. Recall is determined as the number of true positives over the number of true annotations. Precision is the number of true positives over the number of predictions. In Tables 1 and 2, Family Precision refers to pooling all predictions and annotations into one 'family pool' and determining precision accordingly. Recall is also evaluated by pooling. Individual Precision refers to not pooling calls and annotations, but to evaluating precision in each individual separately and taking the average. Genotype Precision is the fraction of (individual) predictions that are (not only true positive, but also) correctly genotyped. In Table 2, we have replaced Genotype Precision by Length Difference, due to that CommonLAW does not genotype and to reflect differences between split-read-driven and insert-size based approaches in terms of breakpoint accuracy.

3.4 Results: real data (platinum genome)

We also evaluated MATE-CLEVER on chromosome 1 of the *platinum genome trio* provided by Illumina, as downloaded from http://www.illumina.com/platinumgenomes, together with the computational annotations, generated through Illumina inhouse-software (Eland and CASSAVA, personal communications with Ole Schulz-Trieglaff, Illumina, available at ftp://ftp.platinumgenomes.org/trio). The platinum trio consists of individuals NA12878 (mother), NA12877 (father) and NA12882 (son).

Table 1. Performance rates for calling and genotyping deletions, using highly stringent evaluation criteria

Coverage	Overall recall 12× / 30×	Family precision 12× / 30×	Individual precision 12× / 30×	Genotype precision 12× / 30×
Length range 10–29 (25 678 true deletions)				
MATE-CLEVER	11.9 / 22.1	90.1 / **90.9**	90.6 / **91.5**	77.2 / 83.4
GATK	57.0 / 69.9	90.9 / 89.8	90.5 / 89.2	**87.0** / **88.3**
PINDEL	**66.3** / **82.7**	**92.4** / 90.4	**93.0** / 91.4	N/A / N/A
Length range 30–49 (3170 true deletions)				
MATE-CLEVER	**55.8** / **69.6**	88.0 / 85.9	88.2 / 86.3	81.9 / 84.6
GATK	15.7 / 23.2	**91.5** / **91.3**	**90.9** / **90.8**	**84.8** / **88.5**
PINDEL	43.0 / 58.8	83.9 / 73.8	86.6 / 78.2	N/A / N/A
Length range 50–99 (1854 true deletions)				
MATE-CLEVER	**48.7** / **55.6**	**77.9** / **74.4**	**78.9** / **74.8**	**78.5** / 76.8
GATK	0.3 / 0.6	66.7 / 73.3	65.0 / 73.7	59.3 / **77.8**
PINDEL	25.0 / 37.2	69.3 / 55.9	72.8 / 61.7	N/A / N/A
Length range 100–249 (1137 true deletions)				
MATE-CLEVER	**34.7** / **42.5**	**73.9** / **63.6**	**76.5** / **64.9**	**77.6** / **70.5**
GATK	0.0 / 0.0	– / –	– / –	– / –
PINDEL	14.1 / 20.1	71.1 / 58.7	75.0 / 63.7	N/A / N/A

Note: See main text for definitions. Best values in each category are typeset in bold face.

Table 2. Performance rates comparing MATE-CLEVER and CommonLAW, using relaxed evaluation criteria

Coverage	Overall recall 12× / 30×	Family precision 12× / 30×	Individual precision 12× / 30×	Length difference 12× / 30×
Length range 10–29 (25 678 true deletions)				
MATE-CLEVER	**13.2** / **23.9**	**93.7** / **93.8**	93.6 / **93.8**	**0.5** / **0.5**
CO.LAW	0.4 / 0.5	93.1 / 89.5	**94.1** / 90.9	17.8 / 16.4
Length range 30–49 (3170 true deletions)				
MATE-CLEVER	**60.7** / **75.2**	93.7 / 92.1	**93.4** / **91.8**	**0.9** / **1.1**
CO.LAW	12.4 / 12.9	**94.2** / **95.0**	92.3 / 88.5	13.6 / 11.0
Length range 50–99 (1854 true deletions)				
MATE-CLEVER	54.1 / 63.1	**87.0** / 85.6	**87.0** / 84.5	**2.3** / **2.8**
CO.LAW	**55.2** / **66.9**	84.3 / **93.6**	77.0 / **89.0**	16.4 / 13.4
Length range 100–249 (1,137 true deletions)				
MATE-CLEVER	38.6 / 48.9	**84.7** / **78.4**	**86.5** / **78.9**	**3.1** / **4.5**
CO.LAW	**39.5** / **52.4**	75.8 / 65.8	77.7 / 68.4	27.9 / 14.8

Note: CommonLAW *cannot genotype*. Because CommonLAW is internal segment size-based, breakpoint predictions are less accurate. See main text for column definitions. Best values in each category are typeset in bold face.

We then aligned the reads using BWA (with the same setting as for the simulated reads) and ran MATE-CLEVER as well as PINDEL on the alignments. Note first that the standard deviation (stddev) of the fragment length distribution is extremely high (≈65 bp), due to the outdated library protocols, which explains the small amount of deletions of length 65–129 bp MATE-CLEVER predicts (We virtually claim that we can discover and genotype indels longer than two times the stddev of the fragment length distribution, which amounts to 30 on current protocols.). See Appendix E for Venn diagrams and statistics, relating MATE-CLEVER, PINDEL and CASSAVA calls. Total amounts of predictions (exclusive calls in parentheses) evaluate as follows (see Appendix E for more details):

– 65–129 bp (1–2 stddev): MATE-CLEVER 69 (22), CASSAVA 130 (79) and PINDEL 256 (195), with 21 shared by all

– 130–194 bp (2–3 stddev): MATE-CLEVER 30 (15), CASSAVA 113 (100), PINDEL 40 (26), with 12 shared by all

– 195–324 bp (3–6 stddev): MATE-CLEVER 87 (22), CASSAVA 45 (28), PINDEL 80 (16), with 16 shared by all

– 325–1000 bp (>6 stddev): MATE-CLEVER 97 (33), CASSAVA 0, PINDEL 157 (93), with 64 shared by the two

Although CASSAVA genotypes can be in conflict with the Mendelian laws (see Appendix A.2), none of the MATE-CLEVER calls are (PINDEL does not genotype).

4 DISCUSSION

Table 1 shows that the GATK's overall recall and precision as well as its genotyping performance are excellent for deletions of 10–29 bp, with recall between 60 and 70%, depending on coverage, and with precision and genotyping rates at ~90%.

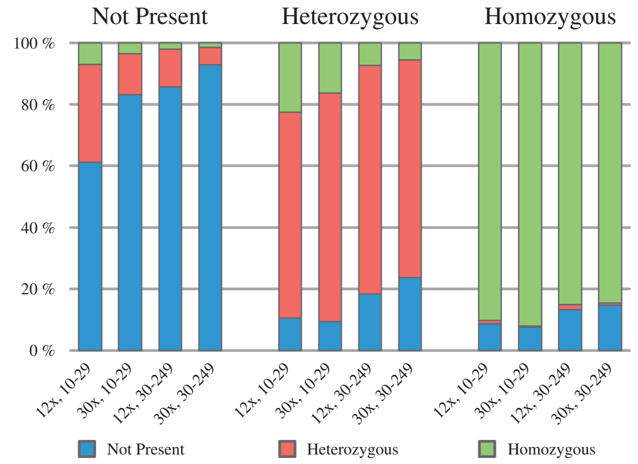

Fig. 2. MATE-CLEVER genotypes, where each class of predictions spans a block of four neighboring bars, are compared with true genotypes, as indicated by colors. Bars refer to different combinations of coverage and indel length. Bar length is measured in percentage. So, e.g. for 30×, 30–249 bp indels predicted as not present (4th bar in 1st block), ~97% are not present (blue fraction) in the respective individual

Table 3. Performance rates when detecting *de novo* deletions are shown

Coverage	Total	Recall	Precision	
			De novo	Inherited
	12× / 30×	12× / 30×	12× / 30×	12× / 30×
Length range 10–29				
MATE-CLEVER	1 / 20	0.0 / 5.7	0.0 / 50.0	100.0 / **35.0**
PINDEL	2221 / 1236	**42.6** / **77.3**	3.3 / 10.8	85.4 / 66.9
GATK	164 / 193	23.3 / 58.5	**25.0** / **52.3**	**63.4** / 38.9
Length range 30–249				
MATE-CLEVER	12 / 47	14.3 / **61.2**	**58.3** / **63.8**	**25.0** / **21.3**
PINDEL	392 / 466	**18.4** / 49.0	2.3 / 5.2	64.8 / 38.4
GATK	8 / 8	4.1 / 10.2	25.0 / 62.5	62.5 / 37.5

Note: Total: Number of *de novo* deletions predicted. Recall: Percentage of true *de novo* deletions that are predicted as such. Precision: Percentage of *de novo* deletion predictions that match true *de novo* deletions. Inherited: Percentage of predicted *de novo* deletions that are true deletions in the child, but are mistyped, i.e. inherited. Best values in each category are typeset in bold face.

PINDEL's discovery performance rates rival those of UG in this size range. However, PINDEL does not allow to genotype. For deletions of 30 bp and longer, MATE-CLEVER has precision and genotyping rates at about those of the GATK (30–49 bp: ≈ 85–88%), but better recall (30–49 bp: 70% for MATE-CLEVER versus 23% for UG, both on 30×).

Table 2 shows that CommonLAW's contribution starts at ~50–60 bp, where it achieves excellent recall and family-pooled precision. Its performance relative to single individuals falls behind MATE-CLEVER and it cannot genotype. Moreover, its accuracy in determining breakpoints is not competitive with that of split-read aligners. Overall, MATE-CLEVER is able to genotype and achieves highly favorable performance rates in terms of assigning and genotyping calls in individuals and in terms of breakpoint accuracy.

Table 3 displays performance rates of MATE-CLEVER, the GATK and PINDEL when discovering *de novo* deletions. For PINDEL, amounts of predictions being typed as 'child only' are seemingly too large—the majority of such calls exist in the child, but are inherited, possibly because *de novo* calling has not yet (personal communication Kai Ye) been implemented in the officially downloadable PINDEL as a special feature. The GATK achieves excellent *de novo* prediction rates on 30× read data, in all size ranges. MATE-CLEVER makes nearly no calls below 30 bp, but outperforms the other tools for deletions larger than 30 bp, keeping stable prediction rates also for 12× data.

As is evident from Figure 2, the majority of MATE-CLEVER calls is correctly typed (as already displayed in Table 1). If MATE-CLEVER mistypes, then it rather over- than under calls deletions. That is, for example, it tends to predict heterozygous calls as homozygous rather than not present, if it fails to genotype correctly.

When running MATE-CLEVER on the platinum trio, one can assume that MATE-CLEVER yields too little predictions <130 bp because this reflects two times the stddev of the insert size distribution (We recall that we claim that we can discover and genotype indels longer than two times the stddev of the fragment length distribution, which amounts to 30 on current protocols.). PINDEL, which, as a split-read mapper, should be much less affected by the large stddev, delivers amounts of predictions, which, in comparison with Venter's genome, are too large. CASSAVA delivers a reasonable amount of predictions. Between 130 and 194 bp, CASSAVA seems to make too many predictions, when relating numbers to the Venter genome. Both MATE-CLEVER and PINDEL deliver reasonable amounts in this size range in this respect. This picture does not change also for deletions longer than 194 bp, for both MATE-CLEVER and PINDEL, but numbers for CASSAVA drastically decrease. Beyond 325 bp (=6 stddev), CASSAVA makes no predictions. Across all size ranges, MATE-CLEVER delivers a substantial fraction of predictions that the other tools do not predict. As a general trend, the agreement between MATE-CLEVER and PINDEL (in relative numbers) is higher than between MATE-CLEVER and CASSAVA or between PINDEL and CASSAVA (see Venn diagrams in Appendix E). Also note that, as discussed in Appendix A.2, the CASSAVA indel calls are often in disagreement with the Mendelian laws of inheritance, which potentially translates into relatively high false-positive prediction rates. Without further wet-lab validations, however, we cannot reach a final conclusion about how to interpret the Venn diagrams in Appendix E.

5 CONCLUSION

We have described a novel combinedly insert size- and split-read-alignment-based (hybrid) approach by which to discover and genotype indels longer than 30 bp. Although the GATK has set the standards for indels of size up to 30 bp, approaches for indels larger than 30 bp had not been available. Here, we close this gap. Our tool MATE-CLEVER discovers and genotypes deletions larger than 30 bp at performance rates that are on a par with those of the GATK for deletions smaller than 30 bp. In doing this, MATE-CLEVER also integrates statistics reflecting the laws of Mendelian inheritance, for enhanced performance rates when dealing with ancestry-related contexts.

We focus exclusively on results for deletions here. With some minor modifications, however, MATE-CLEVER also applies for insertions—both its core engines (Marschall *et al*., 2012, CLEVER); (Marschall and Schönhuth, 2013, LASER) have been designed for also reliably handling insertions (note that the usual limitations owing to read and fragment length do not allow to discover insertions larger than 80 bp). Extrapolating CLEVER's and LASER's performance rates for insertions, which largely agree with those for deletions, may yield reasonable guesses on MATE-CLEVER's performance on insertions.

Still, challenges remain. Neither the GATK nor MATE-CLEVER achieves recall of >70%. Future work will be concerned with raising sensitivity even further, by using improved alignment scores, and also by integrating elements that allow for further improved recalibration of read alignments, such as constructing local haplotypes.

For further results on real data, we refer the interested reader to the GoNL Project (http://www.nlgenome.nl), where MATE-CLEVER contributed to predicting indels in all 231 trios and 19 quartets.

ACKNOWLEDGEMENT

The authors thank various members of the GoNL Consortium and Fereydoun Hormozdiari for fruitful discussions.

Funding: I. Hajirasouliha is supported by the NIH/NHGRI grant (1R01HG005690-01). Hajirasouliha acknowledges funding through the NSERC Canada Graduate Scholarships Michael Smith Foreign Study Supplements Program, which allowed to perform part of this work at CWI, Amsterdam, jointly with the other authors.

Conflict of Interest: none declared.

REFERENCES

Abyzov,A. *et al.* (2011) CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res*., **21**, 974–984.

Albers,C.A. *et al.* (2011) Dindel: accurate indel calls from short-read data. *Genome Res*., **21**, 961–973.

Alkan,C. *et al.* (2011) Genome structural variation discovery and genotyping. *Nat. Rev. Genet*., **12**, 363–376.

Boomsma,D.I. *et al.* (2013) The Genome of the Netherlands: design, and project goals. *Eur. J. Hum. Genet*., [Epub ahead of print, doi:10.1038/ejhg.2013.118, May 29, 2013].

Chen,K. *et al.* (2009) BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods*, **6**, 677–681.

DePristo,M.A. *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet*., **43**, 491–498.

Earl,D. *et al.* (2011) Assemblathon 1: a competitive assessment of *de novo* short read assembly methods. *Genome Res*., **21**, 2224–2241.

Emde,A.-K. *et al.* (2012) Detecting genomic indel variants with exact breakpoints in single- and paired-end sequencing data using SplazerS. *Bioinformatics*, **28**, 619–627.

Gnerre,S. *et al.* (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl Acad. Sci. USA*, **108**, 1513–1518.

Hajirasouliha,I. *et al.* (2010) Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, **26**, 1277–1283.

Hormozdiari,F. *et al.* (2009) Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res*., **19**, 1270–1278.

Hormozdiari,F. *et al.* (2011) Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res*., **21**, 2203–2212.

Jiang,Y. *et al.* (2012) Prism: pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants. *Bioinformatics*, **28**, 2576–2583.

Korbel,J.O. *et al.* (2009) PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*., **10**, R23.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with bowtie 2. *Nat. Methods*, **9**, 357–359.

Lee,S. *et al.* (2009) MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat. Methods*, **6**, 473–474.

Levy,S. *et al.* (2007) The diploid genome sequence of an individual human. *PLoS Biol*., **5**, e254.

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

Li,R. *et al.* (2010) *De novo* assembly of human genomes with massively parallel short read sequencing. *Genome Res*., **20**, 265–272.

Lunter,G. and Goodson,M. (2011) Stampy: a statistical algorithm for sensitive and fast mapping of illumina sequence reads. *Genome Res*., **21**, 936–939.

Marschall,T. and Schönhuth,A. (2013) Sensitive long-indel-aware alignment of sequencing reads. *Tech. Rep*., arXiv:1303.3520.

Marschall,T. *et al.* (2012) CLEVER: clique-enumerating variant finder. *Bioinformatics*, **28**, 2875–2882.

McKenna,A. *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*., **20**, 1297–1303.

Medvedev,P. *et al.* (2010) Detecting copy number variation with mated short reads. *Genome Res*., **20**, 1613–1622.

Quinlan,A.R. *et al.* (2010) Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Res*., **20**, 623–635.

Rausch,T. *et al.* (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**, i333–i339.

Sindi,S. *et al.* (2009) A geometric approach for classification and comparison of structural variants. *Bioinformatics*, **25**, i222–i230.

Sindi,S. *et al.* (2012) An integrative probabilistic model for identification of structural variation in sequencing data. *Genome Biol*., **13**, R22.

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

The International Cancer Genome Consortium. (2010) International network of cancer genome projects. *Nature*, **464**, 993–998.

Ye,K. *et al.* (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.

Zerbino,D. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Res*., **18**, 821–829.

Zhang,J. *et al.* (2012) An improved approach for accurate and efficient calling of structural variations with low-coverage sequence data. *BMC Bioinformatics*, **13** (**Suppl. 6**), S6.