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

Tobias Marschall¹,*, Iman Hajirasouliha² and Alexander Schönhuth¹,*

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

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 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).

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 (Abzyov 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 30bp 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 125bp 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

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

When discovering genetic variations in a trio rather than a single individual, one can take advantage of Mendel’s laws. A variant found originally discovered.

2.2 MATE-CLEVER workflow

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

Step 2: Defining Regions of Interest. At this stage, deletions still may have remained undiscovered in single individuals. So, for each $d \in D_{\text{clever}}$, we declare a window of ±1000bp 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 $\Sigma = \{A, C, G, T\}$ 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 $\Sigma^l \times \Sigma^l$. Correspondingly, we write $R \subseteq \Sigma^l \times \Sigma^l$ for the set of all extracted read pairs, from all individuals.

Step 4: Generating (Split) Alignments with LASER. We align all reads $r \in R$ using our read aligner LASER (Marshall and Schönthuber, 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 $A_i(r)$ to denote the set of alignments of the first and second read end of $r \in R$. We define $A(r) = A_1(r) \cup A_2(r)$ and $A(R) = \bigcup_{r \in R} A(r)$. For each alignment $A \in A(r)$, LASER estimates the probability of it indicating the correct placement of $r$, written $P(A)$, 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 $D_{\text{split}}$ to create a list of candidate deletions $D_{\text{cand}}$ 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 $D_{\text{split}}$ be the set of all deletions supported by split alignments generated in Step 4. We compute the expected support $S_\text{e}(d)$ of a deletion $d \in D_{\text{split}}$ as follows:

$$
S_\text{e}(d) := \sum_{A \in A(R)} \mathbb{I}(A) \cdot P(A)
$$

where $\mathbb{I}(A)$ 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

$$
D_{\text{split}}' := \{d \in D_{\text{split}} : S_\text{e}(d) \geq 0.5\}
$$

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

$$
D_{\text{cand}} := \{d \in D_{\text{split}}' : \text{there exists } d' \in D_{\text{clever}} \text{ such that } \Delta_X(d,d') < T_X \text{ and } \Delta_P(d,d') < T_P \}
$$

where $\Delta_X(d,d')$ is the length difference between $d$ and $d'$ and $\Delta_P(d,d')$ 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 $T_X$ and $T_P$ have to allow for enough flexibility to take that into account without creating too many spurious hits. We found that setting $T_X = 20$, just slightly above the insert size distribution’s stdev, and $T_P = 100$, 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 $P(A)$ 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 $d \in D_{\text{cand}}$, we increase the probabilities $P(A)$ for alignments supporting deletions $d \in D_{\text{cand}}$ as follows.

All deletions $d \in D_{\text{cand}}$ now incur only the minimum phred-scaled cost of 1, whereas all other deletions retain their original costs. To be more precise, a deletion $d' \notin D_{\text{cand}}$ 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 \( P(A) \) for alignments \( A \) indicating deletions \( d \in D_{\text{contd}} \), but not for \( A \) indicating deletions \( d \) not in \( D_{\text{contd}} \). 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 \( D_{\text{contd}} \). We then compute the most likely alignment pair:

\[
(A'_1(r), A'_2(r)) := \arg \max_{(A_1(r), A_2(r))} P(A_1) \cdot P([R(A_1, A_2)]) \cdot P(A_2)
\]

where \( P([R(A_1, A_2)]) \) is the empirical probability (determined from uniquely mappable reads) to observe an internal segment of size \( R(A_1, A_2) \). We discard all other alignments. We also discard alignments where \( P(A) < 1 - 0.001 \), 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 \( \rho_{\text{prior}} \), 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 \( d \in D_{\text{contd}} \) ordered (decreasingly) by expected support \( S_E(d) \). In the following, \( \bar{d}(A) \) be the arithmetic mean of start and end position of \( d \).

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

\[
d \in R(A'_1(r), A'_2(r)) \text{ and } |R(A'_1(r), A'_2(r))| - \mu > |d|/2,
\]

where \( R(A'_1(r), A'_2(r)) \) 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 \( \mu \). Owing to the symmetry of the null distribution \( N_{\mu, \sigma} \), 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

\[
C(d) \in R(A'_1(r), A'_2(r)) \text{ and } |R(A'_1(r), A'_2(r))| - \mu \leq |d|/2.
\]

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 \( p_{\text{FP}}^{\text{ident}} := \int_{|d|=\mu+|d|}^{\mu-|d|} N_{\mu, \sigma}(x)dx \) as the probability of a false positive (a contradicting read being misclassified as supporting) and \( p_{\text{FN}}^{\text{ident}} = p_{\text{FP}}^{\text{ident}} \) as the probability of a false negative (a supporting read misclassified as contradicting).

Let \( n_0 \) and \( n_r \) be the number of supporting and contradicting reads, respectively, and let \( n = n_0 + n_r \). Let \( B_{\rho, p}(k) \) 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)

\[
\begin{align*}
p_{\text{ins}}^{\text{ins}} &\propto B_{\rho, p}(n_0), \quad p = p_{\text{insert}} \\
p_{\text{ins}}^{\text{del}} &\propto B_{\rho, p}(n_0), \quad p = p_{\text{insert}} + 0.5(1 - p_{\text{insert}}) \\
p_{\text{ins}}^{\text{del}} &\propto B_{\rho, p}(n_0), \quad p = 1 - p_{\text{insert}}
\end{align*}
\]

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

\[
n_d = \sum_{r \in R} \mathbb{1}(A'_1(r)) + \mathbb{1}(A'_2(r))
\]

where \( \mathbb{1}(A) \) 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, \( n_0 : n = n_q \) is the number of alignments that contradict \( d \)). We set \( p_{\text{FN}}^{\text{del}} = p_{\text{FP}}^{\text{del}} = 0.01 \). 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 \( n_q \), as obtained here. This yields probabilities \( p_{\text{FN}}^{\text{del}}, p_{\text{FP}}^{\text{del}}, p_{\text{FN}}^{\text{del}} \).

We also consider \( p_{\text{prior}}^{\text{prior}} := 1 - p_{\text{FN}}^{\text{del}}, p_{\text{prior}}^{\text{del}} = p_{\text{prior}}^{\text{prior}} / 2 \), which expresses our prior belief in the zyosity status of deletions. To achieve a high precision, we set \( p_{\text{prior}}^{\text{prior}} = 0.001 \).

Let

\[
p_i := p_{\text{prior}}^{\text{prior}}, p_{\text{ins}}, p_{\text{del}}^{\text{prior}} \quad \text{for } i = 0, 1, 2.
\]

We finally determine the type as

\[
\text{arg max}_{0, 1, 2} \{p_0, p_1, p_2\}
\]

which corresponds to the genotype with the highest posterior probability. Recall that we process deletions \( d \) ordered by expected support \( S_E(d) \). 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 \( T_{\text{de novo}} = 10^{-2} \); we recall that \( p_0 \) is the probability (from Step 7) that a deletion is not present in an individual. All deletions with \( p_0 < T_{\text{de novo}} \) in the child and with \( 1 - p_0 < T_{\text{de novo}} \) 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 \( (p_0, p_1, p_2) \), \( g, h, j \in \{0, 1, 2\} \). For example, \( p_{\text{ins}} = p_{\text{ins}}^g \cdot p_{\text{ins}}^h \cdot p_{\text{ins}}^j \), where we index with \( mn, \text{split, insert} \) 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

\[
q_{ghj} \propto \begin{cases} P(Y_i|X_{ghj}) & \text{if } ghj \text{ respects the Mendelian laws} \\ 0 & \text{otherwise} \end{cases}
\]

See (1) for \( P(Y_i|X_{ghj}) \). We report

\[
\text{arg max}_{ghj} \{q_{ghj} \cdot \rho_{ghj}\}
\]

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 \( \Omega \) be the entire set of annotations. We divide (see section 'Laws of Mendelian inheritance' for notation) \( \Omega = X_{00} \cup X_{11} \cup X_{02} \cup X_{12} \cup X_{22} \) (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 \( \Omega = Y_0 \cup Y_1 \cup Y_2 \) into subsets.

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

\[
(Y_0 \cap X_{10}) \cup (Y_0 \cap X_{11}) \cup \ldots \cup (Y_2 \cap X_{22})
\]

without violating Mendel's laws and such that the resulting call-set 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 12x coverage (which meets the GoNL Project standards) and the other one has 30x coverage. We then aligned all reads with BWA (Li and Durbin, 2009), with default parameters in the 'aln' step and 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 Recall.

| 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).
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%.
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 deleterions here. With some minor modifications, however, MATE-CLEVER also applies for insertions—both its core engines (Marschall et al., 2012, CLEVER); (Marschall and Schönuth, 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


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.