Bioinformatics, 33(24), 2017, 4015–4023
doi: 10.1093/bioinformatics/btx020
Advance Access Publication Date: 7 February 2017
Original Paper

Genome analysis

Genotyping inversions and tandem duplications

Jana Ebler[1], Alexander Schönhuth[2] and Tobias Marschall[1,3,*]

[1]Center for Bioinformatics, Saarland University, Saarbrücken, Germany, [2]Life Sciences Group, Centrum Wiskunde and Informatica, Amsterdam, The Netherlands and [3]Department for Computational Biology and Applied Algorithmics, Max Planck Institute for Informatics, Saarbrücken, Germany

*To whom correspondence should be addressed.

Associate editor: Alfonso Valencia

Received on April 5, 2016; revised on December 8, 2016; editorial decision on January 13, 2017; accepted on January 22, 2017

Abstract

Motivation: Next Generation Sequencing (NGS) has enabled studying structural genomic variants (SVs) such as duplications and inversions in large cohorts. SVs have been shown to play important roles in multiple diseases, including cancer. As costs for NGS continue to decline and variant databases become ever more complete, the relevance of genotyping also SVs from NGS data increases steadily, which is in stark contrast to the lack of tools to do so.

Results: We introduce a novel statistical approach, called DIGTYPER (Duplication and Inversion GenoTYPER), which computes genotype likelihoods for a given inversion or duplication and reports the maximum likelihood genotype. In contrast to purely coverage-based approaches, DIGTYPER uses breakpoint-spanning read pairs as well as split alignments for genotyping, enabling typing also of small events. We tested our approach on simulated and on real data and compared the genotype predictions to those made by DELLY, which discovers SVs and computes genotypes, and SVTyper, a genotyping program used to genotype variants detected by LUMPY. DIGTYPER compares favorable especially for duplications (of all lengths) and for shorter inversions (up to 300 bp). In contrast to DELLY, our approach can genotype SVs from data bases without having to rediscover them.

Availability and Implementation: https://bitbucket.org/jana_ebler/digtyper.git.

Contact: t.marschall@mpi-inf.mpg.de

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

1 Introduction

As of today, several population-scale sequencing projects have been finalized (The 1000 Genomes Project Consortium, 2015; The Genome of the Netherlands Consortium, 2014; The UK10K Consortium, 2015). These projects have revealed an overwhelming amount of new genetic variants and provide the basis to gain deeper insight into the principles of evolution and the association of variants with phenotypes, where disease risks play a particularly important role.

A crucial step in integrating variants into studies on evolution and disease is to genotype and phase them. That is, one has to first determine their zygosity status (genotyping) and then partition the alleles at heterozygous loci into two groups reflecting the two parents of the individual (phasing). The accuracy of the second step crucially hinges on the first step. This implies that one has to operate at utmost accuracy when genotyping—only if genotypes have been determined carefully, variants can finally serve the purposes of downstream studies.

Genotyping variants from next-generation-sequencing (NGS) data, however, can pose involved computational challenges. While genotyping single nucleotide polymorphisms (SNPs) from NGS data already is a routine procedure, genotyping more complex and larger variants is not. Recent advances have pointed out how to do this for shorter (≤ 20–30 bp) insertions and deletions (indels), larger deletions and mobile element insertions (Hehir-Kwa *et al.*, 2016; The 1000 Genomes Project Consortium, 2015). However, the majority of inversions, duplications and translocations are still lacking sound models that allow to determine their genotypes from NGS data. So far, only a few such variants have made their way into phased reference panels (Hehir-Kwa *et al.*, 2016; The 1000 Genomes Project Consortium, 2015). The distinguishing feature of the already phased such variants usually is that they stem from genomic regions that exhibit particularly favorable sequence context. This, however, applies for only little of them. The major part of inversions, duplications and translocations stem from genomic regions that are 'inaccessible', difficult to analyze by short read data. Therefore, simple ad-hoc approaches to genotyping such variants do not work.

In this paper, we provide statistical models and efficient computation schemes that allow to genotype tandem duplications and inversions even though the corresponding read data does not necessarily stem from highly 'accessible' genomic regions. The challenge in this is to control the statistical uncertainties that affect the NGS data that give evidence of such variants. In the first place, aligning the affected reads poses particular difficulties for short read alignment programs such that many of their alignments remain ambiguous. As a result, many likely variant-affected reads are uncertain in terms of their placement. If an alignment is incorrect, that is, the read does not even stem from the variant region in question, it provides no insight into the zygosity status whatsoever. Second, even if correct, one can often interpret an alignment in multiple ways, which may lead to contradicting statements about the existence of variants. The latter case is due to the fact that fragment length can vary and/or that the placement of alignment breakpoints is ambiguous, although the alignment overall indicates the correct placement within the genome, among other issues.

Here, we have been inspired by the models presented by Hehir-Kwa *et al.* (2016, Supplementary Information, Section 3.5), which have led to genotypes of high accuracy for deletions and insertions. These models follow the principle to infer the genotype that is most likely in terms of the read data that supports it. A major problem of such a maximum likelihood (ML) approach for computing genotypes from NGS read data is that a naive evaluation of all relevant reads, together with their uncertainties, results in exponential runtime algorithms. The exponential 'explosion' in runtime is a common problem when taking uncertainties into additional account. Here, for the first time, we present a computation scheme that has runtime linear in the number of reads, even if affected by uncertainties.

An additional advantage of the rigorous statistical approach presented is that the genotype likelihoods computed, that is the probabilities that a variant is absent, heterozygous or homozygous, are highly reliable. Unlike for simple ad-hoc counting strategies, our approach does not get confused by the statistical uncertainties. As usual, the genotype likelihoods can be further used for filtering, thereby controlling the quality level one intends to operate on in downstream analyses, which, in particular, includes computational phasing pipelines. Very often, the underlying data may not allow to correctly distinguish between two genotypes, because the uncertainties affecting the data are too large, or there is too little coverage. In this case, the genotype likelihoods should reflect such situations, such that downstream method can come to the appropriate conclusions.

1.1 Related work

Genotyping single-nucleotide variants (SNVs) and small indels from short-read sequencing data is a routine task for which mature software tools like FreeBayes (Garrison and Marth, 2012) or the GATK (DePristo *et al.*, 2011) are available. Beyond tools for routine whole-genome sequencing (WGS) data, there exist a number of specialized protocols and associated tools. For instance, Stacks (Catchen *et al.*, 2011) targets genotyping of restriction site associated DNA (RAD) markers, while other tools focus on using molecular inversion probe (MIP) sequencing data towards genotyping of short tandem repeats (Carlson *et al.*, 2015) and gene duplications (Nuttle *et al.*, 2013).

Most structural variants (SV) are longer than the sequencing reads and hence cannot be handled by the same approaches as SNVs and short indels. While there is a wealth of tools for SV discovery, there are relatively few approaches that allow to genotype SVs (Lin *et al.*, 2015). Most of these approaches specialize in genotyping insertions and/or deletions of varying lengths. Platypus (Rimmer *et al.*, 2014) generally focuses on smaller indels, but, by making use of local assembly, can also genotype larger ones. Pindel (Ye *et al.*, 2009) also offers a basic procedure for genotyping indels of length up to 50–60 bp. Lumpy (Layer *et al.*, 2014) is an SV discovery tool that also comes with a genotyping module, which is called SVTyper (Chiang *et al.*, 2015). GenomeStrip (Handsaker *et al.*, 2011) has been in use at the 1000 Genomes project (The 1000 Genomes Project Consortium, 2015) for genotyping large deletions. MATE-CLEVER (Marschall *et al.*, 2013) can genotype midsize and long deletions and has been in use at the Genome of the Netherlands project (Hehir-Kwa *et al.*, 2016); the most recent version implements the framework that inspired the present paper.

There are also well-known methods that at least in principle allow to genotype inversions. First, Cortex (Iqbal *et al.*, 2012) generally offers genotyping for various kinds of non-copy-number variants using a colored de Bruijn graph approach, but has not been evaluated for inversions. GASV-PRO (Sindi *et al.*, 2012) offers to discover inversions and a sound statistical model for genotyping variants in general, however, the genotyping option has not been evaluated for inversions.

2 Methods

Our goal is to compute genotypes for given inversions and tandem duplications based on aligned sequencing reads. To this end, we adapt a procedure that we employed previously to genotype insertion and deletions, see Hehir-Kwa *et al.* (2016, Supplement, Section 5.2). For every variant to be genotyped, we consider all reads from the corresponding region. This set of reads, referred to as $\mathcal{R}$, is used to determine the genotype for the respective variant. The goal is to compute probabilities for the three possible genotypes G_i, for $i = 0, 1, 2$, given all reads $\mathcal{R}$, where G_0, G_1, G_2 represent that the variant in question is absent, heterozygous or homozygous, respectively. We use the reads from $\mathcal{R}$ to compute a posterior probability for each genotype. Finally the genotype for which the probability is highest is the result.

For each read $R \in \mathcal{R}$, let $\mathbb{P}(A^+(R))$ denote the probability that its alignment is correct and let $\mathbb{P}(A^-(R))$ denote the probability that it is not. Then, $\mathbb{P}(G_i|A^+(R))$ is the probability for genotype G_i under the assumption that the read is mapped correctly and $\mathbb{P}(G_i|A^-(R))$, the probability for a genotype under the assumption that the alignment of the read is wrong. Using Bayes' theorem and the assumption of constant priors over the genotypes, the probability for a particular genotype given all reads can be expressed as follows (Hehir-Kwa *et al.*, 2016):

$$\begin{aligned}\mathbb{P}(G_i|\mathcal{R}) \propto \prod_{R\in\mathcal{R}} \Big[&\mathbb{P}(A^+(R))\mathbb{P}(G_i|A^+(R)) \\ &+(1-\mathbb{P}(A^+(R)))\mathbb{P}(G_i|A^-(R))\Big].\end{aligned} \tag{1}$$

Note that this expression can be evaluated in time linear in $|\mathcal{R}|$, and hence avoids to explore all $2^{|\mathcal{R}|}$ possible combinations of correctly/wrongly mapped reads, which would result in exponential runtime. In the following, we derive the terms needed to evaluate (1). We set

$$\mathbb{P}(G_i|A^-(R)) = \mathbb{P}(G_i),$$

where $\mathbb{P}(G_i)$ expresses our prior belief in genotype G_i, because if the considered read does not stem from the region it does not give any information about the genotype.

The probabilities for the alignment of the read to be correct, $\mathbb{P}(A^+(R))$, and to be incorrect, $\mathbb{P}(A^-(R))$, can be obtained as follows.

$$\mathbb{P}(A^-(R)) = max\{0.05, p_{wrong1} \cdot p_{wrong2} + (1 - p_{wrong1}) \cdot p_{wrong2} + p_{wrong1} \cdot (1 - p_{wrong2})\} \tag{2}$$

where

$$p_{wrong1} = 10^{-\frac{MapQuality(left)}{10}}$$

$$p_{wrong2} = 10^{-\frac{MapQuality(right)}{10}}$$

are the probabilities for the two read ends of a read pair not to be mapped correctly, so p_{wrong1} gives $\mathbb{P}(A^-(R))$ for the left end and analogously p_{wrong2} for the right end of the considered read pair. The *MapQuality* of a read is directly taken from the input BAM file. By considering the maximum in Equation (2), we ensure to never fully 'trust' a read alignment, and hence account for mapping uncertainties not recognized by the read aligner. Finally $\mathbb{P}(A^+(R))$ is computed as $1 - \mathbb{P}(A^-(R))$.

The only yet unspecified probability needed to evaluate Equation (1) is $\mathbb{P}(G_i|A^+(R))$. While we have studied these quantities before for genotyping deletions (Hehir-Kwa *et al.*, 2016), we develop them here for inversions and duplications. These two cases are treated separately in the following sections. In both cases, the central idea is the same: for each read, we compute the probability $\mathbb{P}(G_i|A^+(R))$ for all three genotypes G_0, G_1, and G_2, which tells us how likely each genotype is given only this one read R. The intuition is that each individual read gives rise to a rather flat, yet not uniform, genotype distribution; that is, a single read only makes a weak statement about genotypes. To become more confident in a genotype, reflected by a more pronounced difference between genotype probabilites, many reads need to be combined [according to Equation (1)].

2.1 Approach for inversions

To compute $\mathbb{P}(G_i|A^+(R))$ for inversions, two cases must be distinguished, depending on the positions of the mapped reads in the dataset. In both cases, reads supporting the variant and reads supporting the reference allele are taken into consideration for computing the likeliest genotype.

2.1.1 Reversed read evidence

We first consider cases where one end of the paired-end read is mapped outside of the inversion and the other end is mapped completely inside the inversion. Under the assumption of no alignment uncertainty, a read stems from the inversion allele if and only if the orientation of the read end located inside the inversion is the same as for the other read end. This means the read end in the inversion was reversed when being mapped to the reference. See Figure 1 for an example. Such a read supports the presence of the considered inversion in the sequence. Let R_1 be a read that supports the inversion.

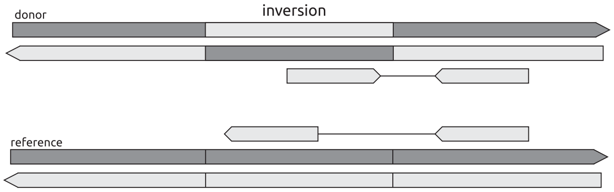


Fig. 1. Reversed read evidence. Above the original read is shown and below the read is mapped to the reference. Note that the end that stems from in between the inversion breakpoints changes its orientation when being mapped to the reference

According to Bayes' theorem and the assumption of constant priors, it holds that $\mathbb{P}(G_i|A^+(R_1)) \propto \mathbb{P}(A^+(R_1)|G_i)$. The latter term can be computed as follows. $\mathbb{P}(A^+(R_1)|G_0) = 0$, because if the inversion is absent the read cannot stem from the region. Then $\mathbb{P}(A^+(R_1)|G_1) = \frac{1}{2}[\mathbb{P}(A^+(R_1)|G_0) + \mathbb{P}(A^+(R_1)|G_2)]$, reflecting the case that one randomly picks one of the two chromosomal copies with only one containing the inversion and then generates the read from it. These considerations lead to the expression

$$\mathbb{P}(G_i|A^+(R_1)) = \begin{cases} 0 & \text{if } i = 0, \\ \dfrac{1}{3} & \text{if } i = 1, \\ \dfrac{2}{3} & \text{if } i = 2. \end{cases} \tag{3}$$

If the orientation of the read mapped in between the inversion breakpoints is not changed, the read must stem from a sequence without the inversion. Let R_2 be such a read. This case is treated analogously to the previous one, but this time we have $\mathbb{P}(A^+(R_2)|G_2) = 0$, leading to

$$\mathbb{P}(G_i|A^+(R_2)) = \begin{cases} \dfrac{2}{3} & \text{if } i = 0, \\ \dfrac{1}{3} & \text{if } i = 1, \\ 0 & \text{if } i = 2. \end{cases} \tag{4}$$

2.1.2 Split read evidence

Read pairs for which one read end stretches across one of the inversion breakpoints cannot be mapped by standard read mappers. To leverage these reads for genotyping, we extended the read mapper LASER (Marschall and Schönhuth, 2013) to detect inversion-type split alignments (Fig. 2). While other tools exist (such as BWA MEM, Li, 2013) that are, in principle, able to detect such splits, we chose to extend LASER because it is specifically geared towards sensitivity, which is needed to compute meaningful mapping qualities. Since LASER is based on partial banded alignments that extend seed hits, implementing this feature only required to combine anchor alignments of opposite directionality (showcasing the power of this technique). When using option --inversions, LASER outputs these split reads encoded as IV tags in the BAM output.

For genotyping, we evaluate these tags and decide whether a (split) alignment across an inversion breakpoint supports the inversion or the reference. After being mapped to the reference sequence, the part of this read end located inside the inversion breakpoints $(:= A)$ is reversed, while the other part outside the breakpoints $(:= B)$ is not, as illustrated in Figure 2. The distance between the alignments of A and B equals $length(inversion) - length(A)$. To support the inversion, the following requirements have to be fulfilled besides the reversed orientation of A: One end of part B must agree with the inversion breakpoint the read stretches over and one end of part A has to agree with the other inversion breakpoint. The read supports the inversion if and only if these requirements are fulfilled. Otherwise it supports the reference sequence. Just as in Section 2.1.1, the probability $\mathbb{P}(G_i|A^+(R))$ for the genotype G_i is computed using Equation (3) and Equation (4) for reads supporting inversion allele and reference allele, respectively.

For split-reads $\mathbb{P}(A^-(R))$ is computed as

$$\mathbb{P}(A^-(R)) = max\left\{0.05, 10^{-\frac{MapQuality(read)}{10}}\right\} \tag{5}$$

Here *read* describes the read end which stretches over the breakpoint. Since for split reads only these read ends are considered, only the probability that such an end is wrong is taken into account. Again, $\mathbb{P}(A^+(R))$ is computed as $1 - \mathbb{P}(A^-(R))$.

2.2 Approach for duplications

To genotype duplications, we use a statistical framework that considers all read pairs with at least one read end aligned to lie completely inside the duplication. Figure 3 shows how such read pairs can be placed on the originating duplication allele and what the resulting alignments look like. Compared to inversions, we now have to overcome an additional complication: No read pair gives direct evidence of the reference allele. All read pairs that originated from the reference allele could potentially also have originated from the duplication allele. Read types B and D in Figure 3 are examples of this. For inversions, that was not the case and we could restrict our attention to read pairs that can uniquely be determined to either stem from the variant allele or from the reference allele.

In the following, reads that unambiguously support a duplication are denoted as *supporting* and read pairs that can have originated from both alleles (reference/duplication) are denoted as *neutral*. In order to genotype duplications, the main idea is to consider the proportion of supporting and neutral reads, which can be achieved within the same framework as for inversions. Again, our goal is to compute $\mathbb{P}(G_i|A^+(R))$ in order to evaluate Equation (1).

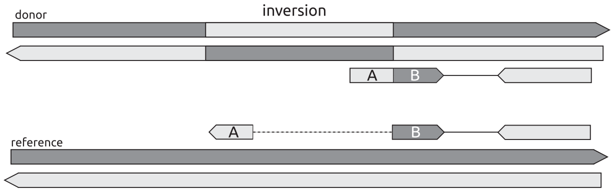


Fig. 2. Split Read Evidence. The light part of the left read end is reversed after mapping while the rest is not changed. There will be a long gap between the light and the dark part of the left read end

We will approach short and long duplications separately since they are qualitatively different in terms of read types shown in Figure 3. For short duplications, read pairs of types A, B, C, D, and E exist. Read pairs of type F do not exist when the duplication is smaller than the fragment size (i.e. insert size plus read lengths). For long duplications, read pairs of type F are present but types A and C do not exist. Note that read pairs of type E exist for short and for long duplications, but their relative placement on the reference genome depends on the duplication length. For a duplication longer than the fragment size, the order of forward and backward read ends on the reference gets reversed as can be seen in 5. We give a precise definition of 'short' and 'long' after introducing some notation in the following section.

2.2.1 Insert size evidence

For duplications, the distance of the alignments of the two read ends in a read pair can be leveraged for genotyping. Consider a scenario where a read pair stems from the duplication allele and the left read end lies outside and the right read end lies inside the duplication. The right read end could have originated from the first copy or the second copy of the duplication in the donor genome. In the latter case, we observe a reduced distance of the aligned read pair, as illustrated in Figure 3, read type A. In slight abuse of terminology, we refer to the distance of the two aligned reads as *insert size*, depicted in Figure 4. The insert size is computed by subtracting the end position of the read end which is mapped to the forward strand from the start position of the read end which maps to the reverse strand. In case the orientations of the read ends are reversed, this value is negative (Fig. 4).

We adopt the common assumption (Marschall *et al.*, 2012) that the insert size follows a normal distribution under the null hypothesis (i.e., when the read pair stems from the reference allele and has been mapped correctly). Mean and standard deviation of this distribution can be robustly estimated from the aligned reads (Marschall *et al.*, 2012). We hence assume these quantities to be known and denote them as μ and σ in the following. A normal distribution with this mean and standard deviation is written as $\mathcal{N}_{\mu,\sigma^2}$.

Let ℓ denote the length of the duplication to be genotyped, which we define as the length of the repeat unit that is duplicated and therefore occurs twice in the genome. Read pairs spanning a copy of the duplicated sequence, such as read pairs A and C in Figure 3, will have an observed insert size distributed according to $\mathcal{N}_{\mu-\ell,\sigma^2}$, whereas read pairs from the reference allele exhibit insert sizes distributed according to $\mathcal{N}_{\mu,\sigma^2}$.

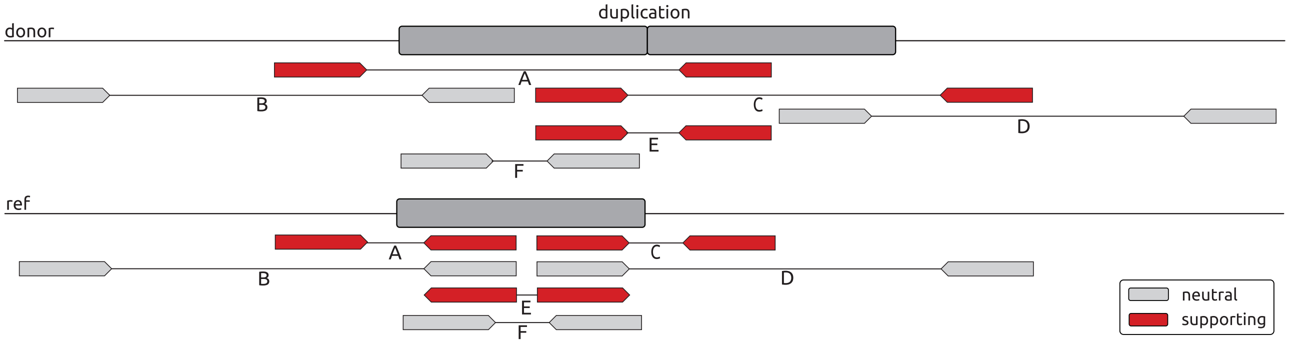


Fig. 3. Different types of read pairs originating from a duplication allele and how they are mapped to the reference sequence. Thick horizontal lines indicate donor and reference genomes, with the duplication shown as gray boxes. Mapping *supporting* read pairs (shown in red; A, C and E) to the reference gives rise to shorter observed insert size, while the insert size of *neutral* read pairs (shown in gray, B, D and F) agrees with the null distribution $\mathcal{N}_{\mu,\sigma^2}$

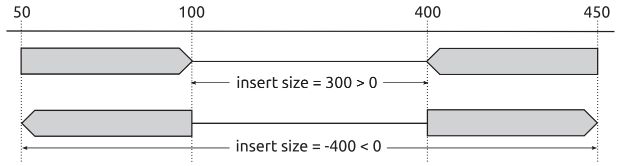
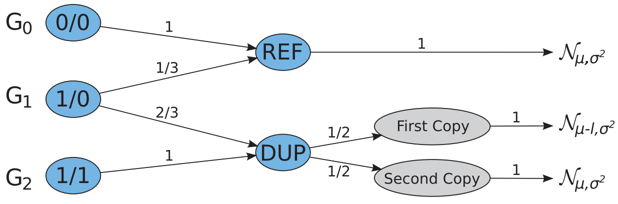


Fig. 4. Definition of the insert size of a paired-end read



G_0
0/0
G_1
1/0
G_2
1/1
1
1/3
2/3
1
REF
DUP
1
1/2
1/2
First Copy
Second Copy
1
1
$\mathcal{N}_{\mu,\sigma^2}$
$\mathcal{N}_{\mu\text{-}l,\sigma^2}$
$\mathcal{N}_{\mu,\sigma^2}$

Fig. 5. Link between genotypes (left column) and distributions of an observed insert size (right column) mediated by the allele the read pair stems from (middle, blue) and, in case of the duplication allele, by the copy of the duplication the anchor read originates from (middle, gray). We show the scenario where the left read end is the anchor; if the right read end is the anchor, the roles of 'first copy' and 'second copy' are swapped. Edge labels indicate probabilities

2.2.2 Short duplications

We call a duplication *short* when its length ℓ is smaller than $\mu + len(ReadEnd)$. This implies that read pairs of type F (Fig. 3) are very unlikely to exist.

The most important evidence for short duplications comes from read types A, B, C, and D. They are characterized by one read end being aligned completely inside the duplication and the other read end being aligned (at least partially) outside the duplication. We call the read completely inside the duplication *anchor* read. As outlined above, these read pairs can either lead to an observed insert size distribution of $\mathcal{N}_{\mu,\sigma^2}$ (for types B and D) or of $\mathcal{N}_{\mu-\ell,\sigma^2}$ (for types A and C). To derive $\mathbb{P}(G_i|A^+(R))$, we consider how these read pairs can be generated, as illustrated in Figure 5. The probability of whether a given read pair has originated from the reference or from the duplication allele obviously depends on the genotype. While they equal 1 for homozygous genotypes, a heterozygous genotype leads to probabilities to stem from the reference allele or duplication allele of 1/3 and 2/3, respectively. The duplication allele is twice as likely because the anchor read is completely inside the duplicated region which exists twice on the duplication allele. In case a read came from the reference allele, we observe an insert size distribution of $\mathcal{N}_{\mu,\sigma^2}$. In case it came from the duplication allele, two scenarios are possible: either the anchor read originated from its first copy or from it originated from its second copy, leading to either an observed insert size distribution of $\mathcal{N}_{\mu,\sigma^2}$ or of $\mathcal{N}_{\mu-\ell,\sigma^2}$. The whole process is illustrated in Figure 5.

Each path from left to right in Figure 5 contributes to the probability that a given genotype in the left column gives rise to read pairs with the observed insert size distribution given in the right column. By summing up all paths for each genotype, we obtain

$$\mathbb{P}(G_i|A^+(R)) := \begin{cases} \dfrac{1}{Z}\left(1 \cdot \mathcal{N}_{\mu,\sigma^2}(i_R)\right) & i = 0, \\[2ex] \dfrac{1}{Z}\left(\dfrac{2}{3} \cdot \mathcal{N}_{\mu,\sigma^2}(i_R) + \dfrac{1}{3} \cdot \mathcal{N}_{\mu-l,\sigma^2}(i_R)\right) & i = 1, \\[2ex] \dfrac{1}{Z}\left(\dfrac{1}{2} \cdot \mathcal{N}_{\mu,\sigma^2}(i_R) + \dfrac{1}{2} \cdot \mathcal{N}_{\mu-l,\sigma^2}(i_R)\right) & i = 2, \end{cases} \tag{6}$$

where i_R denotes the insert size observed for read R and Z is a normalization factor given by

$$\begin{aligned} Z := \mathcal{N}_{\mu,\sigma^2}(i_R) + \left(\frac{2}{3} \cdot \mathcal{N}_{\mu,\sigma^2}(i_R) + \frac{1}{3} \cdot \mathcal{N}_{\mu-l,\sigma^2}(i_R)\right) \\ + \left(\frac{1}{2} \cdot \mathcal{N}_{\mu,\sigma^2}(i_R) + \frac{1}{2} \cdot \mathcal{N}_{\mu-l,\sigma^2}(i_R)\right). \end{aligned} \tag{7}$$

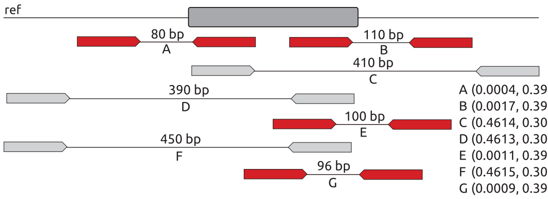
Read pairs of type E are processed using the same formula. Summing over all read pairs by plugging Equation (6) into Equation (1) yields the sought genotype likelihoods.

2.2.3 Long duplications

We call a duplication *long* when its length ℓ is longer than or equal to $\mu + len(ReadEnd)$. This makes it possible that read pairs of type F exist, which require some extra attention. One observes twice the number of such read pairs per duplication allele than per reference allele and their expected number grows linearly with the duplication length (in fact constituting the signal that coverage-based copy number estimation tools use). Like for short duplications, we seek to only use those read pairs that span a duplication breakpoint, either start or end of the duplicated region or the internal breakpoint between the two copies of the duplication. That is, we want to use read pairs of type E but exclude those of type F. For long duplications, however, this distinction can be made with very good accuracy based on the observed insert size [for values of μ, σ, and $len(ReadEnd)$ common in practice]. Read pairs of type E have reversed orientations or at least they overlap after they have been mapped to the reference. After discarding read pairs of type F, the genotyping proceeds in the same way as for short duplications. The product over all reads $R \in \mathcal{R}$ in Equation (1) runs over all read pairs that have an anchor read mapped completely inside the duplication. It is important to note that read pairs of type E have *two* anchor reads and hence need to be counted twice in this product. This ensures that the expected number of counted read pairs of type E equals the expected number of read pairs of types B and D. Note that observing reads of types A or C becomes increasingly unlikely for growing duplication lengths.

2.3 From likelihoods to genotypes

To illustrate our genotyping algorithm, we consider the example shown in Figure 6. It shows reads supporting a duplication (in red) and neutral ones (in gray) mapped to a reference genome. We assume the duplication to be 300 bp long and further assume to have observed the insert sizes shown in Figure 6. In this example, we set $\mathbb{P}(A^+(R)) = 0.99$ for all reads and we assume a null distribution with $\mu = 400$ and $\sigma = 80$. The genotype probabilities can now be computed by plugging Equation (6) into the likelihood function (1). Using the parameters mentioned above, this results in the following genotype probabilities.



ref
80 bp
A
110 bp
B
410 bp
C
390 bp
D
100 bp
E
450 bp
F
96 bp
G
A (0.0004, 0.3999, 0.5997)
B (0.0017, 0.3997, 0.5986)
C (0.4614, 0.3077, 0.2309)
D (0.4613, 0.3077, 0.2310)
E (0.0011, 0.3998, 0.5992)
F (0.4615, 0.3077, 0.2308)
G (0.0009, 0.3998, 0.5993)

Fig. 6. Example of reads covering a duplication. Supporting reads are shown in red, neutral ones in gray. The gray box illustrates the region which is duplicated in the donor genome. Furthermore, the assumed insert sizes and corresponding probabilities $(\mathbb{P}(G_0|A^+(R)), \mathbb{P}(G_1|A^+(R)), \mathbb{P}(G_2|A^+(R)))$ for absent, heterozygous and homozygous genotypes, are shown

$$\mathbb{P}(G_0|\mathcal{R}) = \prod_{R\in\mathcal{R}} 0.99\cdot\mathcal{N}_{400,80^2}(i_R)+\frac{0.01}{3}$$

$$\mathbb{P}(G_1|\mathcal{R}) = \prod_{R\in\mathcal{R}} 0.99\cdot\left(\frac{2}{3}\cdot\mathcal{N}_{400,80^2}(i_R)+\frac{1}{3}\cdot\mathcal{N}_{100,80^2}(i_R)\right)+\frac{0.01}{3}$$

$$\mathbb{P}(G_2|\mathcal{R}) = \prod_{R\in\mathcal{R}} 0.99\cdot\left(\frac{1}{2}\cdot\mathcal{N}_{400,80^2}(i_R)+\frac{1}{2}\cdot\mathcal{N}_{100,80^2}(i_R)\right)+\frac{0.01}{3}$$

This, after normalization, results in the probabilities $(\mathbb{P}(G_0|\mathcal{R}), \mathbb{P}(G_1|\mathcal{R}), \mathbb{P}(G_2|\mathcal{R})) = (0.004, 0.399, 0.597)^T$, which indicates that the homozygous genotype is the likeliest.

After genotype likelihoods have been computed as explained above and illustrated in the example, the likeliest genotype is reported as result. DIGTYPER also outputs genotype likelihoods as phred-scaled posterior probabilities when writing a VCF file. That is, $-10\cdot\log_{10}(p_i)$ is reported as genotype likelihood for genotype i with posterior probability p_i. The difference of the phred-scaled posterior of the likeliest and second-likeliest genotype is used to decide whether to report a genotype at all or "./." to indicate too large ambiguity; the default threshold for this difference is set to 20. When this filter has not been passed but the phred-scaled posterior for genotype *0/0* (homozygous in the reference allele) is below a user-specifiable threshold (default set to 20), then genotype *1/.* is reported to indicate that at least one alternative allele is present, i.e. the genotype is believed to be either *1/0* or *1/1* but the data is insufficient to distinguish these two cases (as in the above example).

3 Results

We evaluated DIGTYPER on simulated data and on a real data set provided by the *Genome in a Bottle Consortium (GIAB)* (Zook *et al.*, 2015). We compared our algorithm to DELLY (Rausch *et al.*, 2012), Pindel (Ye *et al.*, 2009) and SVTyper (Chiang *et al.*, 2015). While SVTyper can genotype given calls, DELLY and Pindel cannot; that is, they can only genotype their own discoveries, which explains why we can only evaluate DELLY and Pindel on their own calls in the following. For head-to-head comparisons with DELLY and Pindel, we evaluate our own method as well as SVTyper on only DELLY/Pindel calls.

As one can expect to see variant databases being steadily filled with inversions and tandem duplications in the short and midterm future, there is an obvious need for tools that do not depend on their own discovery functionalities. Therefore, we will further also evaluate our own method and SVTyper on all variants known relative to the respective evaluation scenario so as to gauge the extent of variants that can be genotyped by a discovery-independent approach.

3.1 Simulated data

For generating simulated data, we used the reference sequence of human Chromosome 1 (version: hs37d5). We inserted inversions and duplications of varying lengths into this chromosome, with neighboring inserted variants separated by one million bp of reference sequence. We then used SimSeq (Earl *et al.*, 2011) for generating read data from the resulting sequence. The mean μ of the length of the generated fragments was 550 bp, at a standard deviation σ of 140. Read ends were of length 148 bp. This mimics the parameters from the real GIAB dataset, so as to have a realistic simulation scenario. We used *bwa* (Li and Durbin, 2009) to map the reads to the reference sequence and to create BAM files as input for the programs. We varied the coverage and obtained datasets at coverages of 4×, 12× and 60×, all of which reflect realistic settings. While the length of the simulated inversions were 100, 300, 500 and 800 bp, the length of the duplications was set to 200, 300, 500 and 800bp—because duplications shorter than read length cannot be detected. Reads reflecting heterozygous variants were generated by simulating reads from both our simulated sequence and the reference Chromosome 1, which were subsequently merged using SAMtools (Li *et al.*, 2009).

We then ran DELLY and Pindel on the generated datasets in discovery mode to generate inversion and duplication calls. The positions of these calls were then provided as input to the two genotyping programs. Only predictions tagged as 'precise' by DELLY were considered, since for split read analysis the breakpoints should be accurate. In our evaluation experiments, we only considered DELLY and Pindel variants whose center points were found to not deviate by more than 50 bp from the true center points. On these variants, we compare our genotype predictions ['DIGTYPER (retype)' in Fig. 7 and Supplementary Fig. S1] and those of SVTyper ['SVTYPER (retype)' in Fig. 7 and Supplementary Fig. S1] with the ones from *DELLY* and Pindel. Additionally, we also evaluate our program and SVTyper on all variants we have inserted in our simulated data, which DELLY and Pindel do not allow to do ('DIGTYPER' and 'SVTYPER' in Fig. 7 and Supplementary Fig. S1).

3.1.1 Results for inversions

Figure 7 (top) shows the results of DELLY, SVTyper and DIGTYPER for inversions. The Pindel results can be found in Supplementary Figure S1.

Genotyping DELLY variants. DELLY could only discover a small amount of inversions of length 100 bp and was not able to give genotype predictions for the majority of those. While SVTyper gave false genotype predictions for more than 75% of the variants detected by DELLY, DIGTYPER genotyped almost all of them correctly, with only about 1% false predictions. For inversions of length 300 bp, DELLY detected between 80% and 90% of the variants at coverages 12× and 60×, and genotyped almost all correctly (only about 1% of errors). The results of DIGTYPER for these variants are almost identical, with slightly fewer false predictions. In contrast, SVTyper yielded much more false genotypes for inversions of length 300 bp. At coverage 12×, between 20% and 25% of the genotype predictions were wrong. In an overall account, our method, SVTyper and DELLY largely agreed on all other length ranges (starting from 500 bp) and coverage rates (starting from 12×) and genotyped between 98% and 100% of all discovered variants correctly, with DELLY and SVTyper yielding slightly more false predictions in comparison to DIGTYPER. For variants longer than 300 bp, DIGTYPER did not make a single wrong genotype call.

Genotyping inserted variants. Next, we compared performance of DIGTYPER and SVTyper when re-genotyping the known inversions we had implanted into Chromosome 1.

For inversions of 300 bp and longer, DIGTYPER was able to correctly genotype 60–80% of those already at coverage 4×, while less than 2% were wrongly genotyped. For the remaining inversions (fraction missing to 100%), DIGTYPER either refused to genotype for the lack of enough evidence or correctly reported the inversions to have at least one alternative allele ('1/.' calls, dark brown in Fig. 7). At coverages of 12× and 60×, more than 80% for inversions

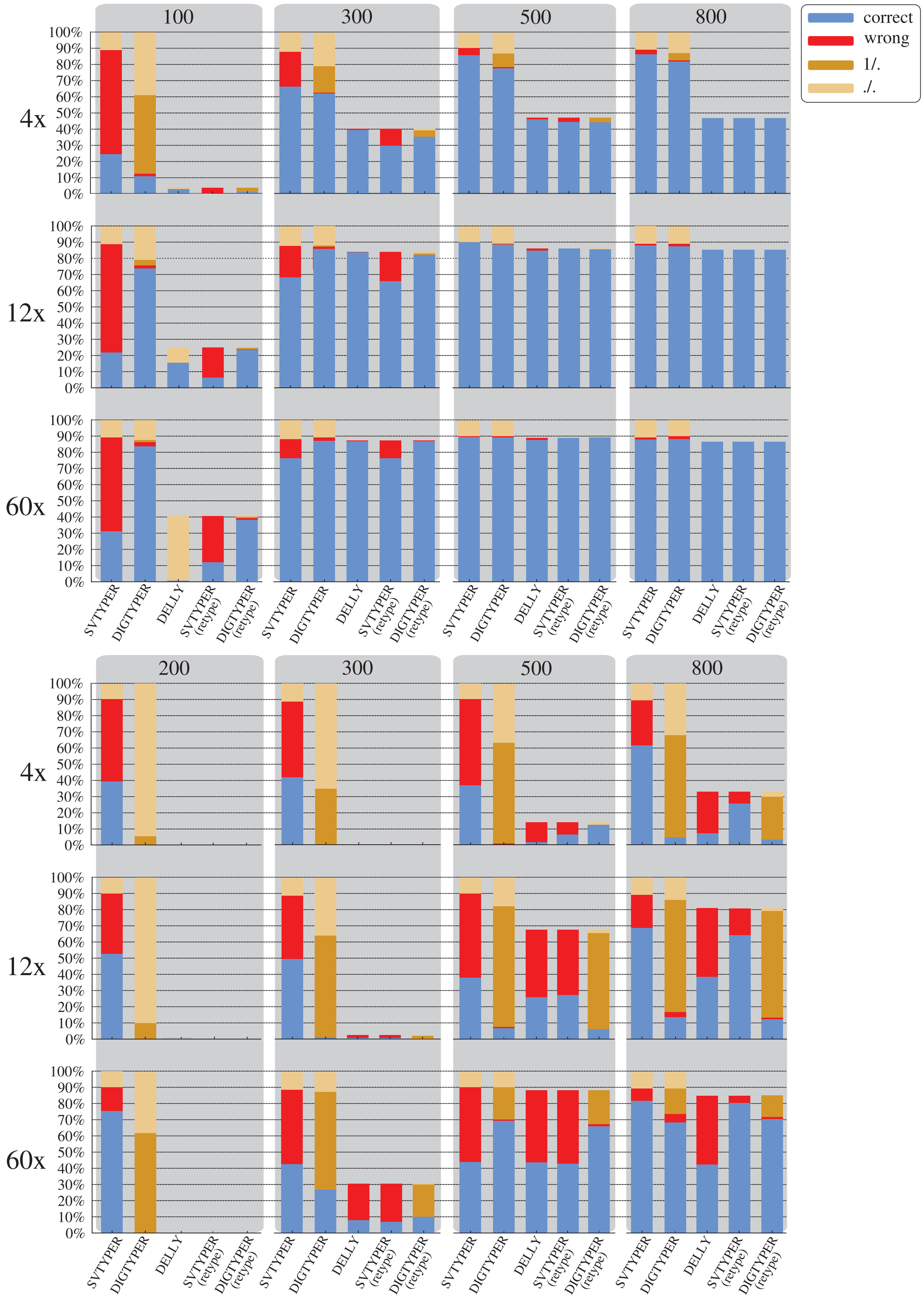


Fig. 7. Results on simulated data for inversions (top) and duplications (bottom). The panels show the genotype predictions of SVTyper and DIGTYPER for all inserted variants and for the ones found by DELLY ['DIGTYPER (retype)', 'SVTYPER (retype)']. Results are stratified by length (gray columns, length given at the top of each column) and by coverage (rows). Light brown (./.) refers to cases where a tool did not make a genotype call and dark brown (1/.) refers to cases where DIGTYPER correctly reported the presence of at least one alternative allele (without resolving heterozygous versus homozygous) (Color version of this figure is available at *Bioinformatics* online.)

of all length ranges were typed correctly with, again less than 3% error rates. In concordance with our expectation, short inversions (100 bp) at low coverage (4×) turned out to be hardest to genotype; only 10% were typed correctly, about 48% were (correctly) reported as '1/.' and 40% were not typed, while about 2% were wrongly typed. By making use of our statistical machinery, we could hence decide correctly in 48% of the cases that the variant existed (1/.) while refusing to distinguish between heterozygous and homozygous, since the data uncertainties were too high to allow us to do so. In general, DIGTYPER yielded very few errors, even at low coverage. This is enabled by correctly refusing to type based on the data given, and to issue genotypes *1/.* or *./.* in that case.

Running SVTyper on all true inversions inserted into the reference gives false genotype predictions for 60–70% of all inversions of length 100 bp at all coverages. This points out a significant advantage of our tool, as it thus establishes the first approach to genotype short inversions with high accuracy, in particular for coverages of at least 12×. While for inversions of size 300 bp SVTyper still yields a high amount of errors, with false genotype predictions of 10–30% at all coverages, SVTyper and DIGTYPER performed similarly for longer variants and, at each coverage, genotyped between 80% and 90% of the inversions correctly.

3.1.2 Results for duplications

The results for duplications are shown in Figure 7 (bottom). Again those of Pindel can be found in Supplemental Figure S1.

Genotyping DELLY variants. DELLY was able to discover significant fractions of implanted duplications only for lengths 500 bp and 800 bp. Comparing genotyping performance for 200 bp was hence not possible. For duplications of length 300 bp, a comparison was only possible at 60×, where DELLY detected 30% of the inserted variants. Only one third of those could be genotyped correctly by DELLY. Also SVTyper could not give reliable genotype predictions for short duplications up to 500 bp since it genotyped most variants as heterozygous. This caused similar amounts of false predictions as DELLY. These length classes are particularly challenging since only few anchor reads, which lie completely inside the duplication exist. Furthermore, such duplications are also relatively short compared to the standard deviation of the insert sizes ($\sigma = 140$), inducing uncertainty when distinguishing supporting from neutral read pairs. DIGTYPER recognized this uncertainty and reported many *./.* and *1/.* genotypes. For longer duplications or higher coverage, the uncertainty decreased, as expected. Even for longer duplications, DELLY was not able to distinguish homozygous and heterozygous tandem duplications, since almost all variants found were reported as *1/0*. This leads to a very high number of false predictions, as evidenced by the red bars in Figure 7 (bottom). Only at high coverage, larger amounts of duplications of length 800 bp could be genotyped correctly by SVTyper (about 90% of all detected variants). At high coverage, DIGTYPER was able to genotype about 75% of the detected variants (500 bp and longer) correctly, while for the rest, it mostly reported *1/.* genotypes and only about 2% of false predictions.

Genotyping of inserted variants. Again, we used SVTyper and DIGTYPER to genotype the known variants inserted into the reference, which mimics the application of genotyping data base variants. Compared to the results for inversions, DIGTYPER genotyped a larger fraction of duplications as *1/.*, especially at coverages 4× and 12×, indicating that at least one chromosome carries the variant with a high probability. This reflects the fact that duplications are fundamentally more difficult to genotype than inversions, because read pairs that give direct evidence for the reference allele do not exist. SVTyper shows a similar behavior as for the DELLY variants, resulting in very high amounts of false predictions for short variants. Only for long duplications (800 bp), SVTyper genotypes many more variants correctly, but still at significantly higher error rates than DIGTYPER.

3.2 *GIAB* data

We used an Illumina HiSeq dataset from the *Genome in a Bottle Consortium* (Zook *et al.*, 2015) for individual HG003 of the Ashkenazi trio, which was sequencing to 60× coverage. Since we lacked reliable ground truth data of genotyped inversions and duplications, we conceived the following experiment. First, we ran DELLY to discover and genotype inversions and duplications on this data set. Second, we considered all inversions and duplications reported by The 1000 Genomes Project Consortium (2015) and The Genome of the Netherlands Consortium (2014), which we refer to as *data base variants*. Note that the Ashkenazi trio is not part of either of the two projects. We genotyped these data base variants in the GIAB individual with DIGTYPER and with SVTyper. Then, we determined for each data base variant whether it matched at least one variant discovered by DELLY (with a center point distance and length difference of up to 200 bp). Next, we determined the intersection between the sets of data base variants typed *1/1* or *1/0* or *1/.* by DIGTYPER or SVTyper on the one hand and the set of data base variants matching a DELLY call on the other hand. The results are shown in Figure 8. Still lacking a ground truth, we can now compare DELLY calls to DIGTYPER and SVTyper predictions. Since DELLY often reports multiple overlapping predictions (sometimes with different genotypes) matching the same data base variant, we did not compare genotypes, but only absence/presence signals. We want to emphasize that data base variant typed as *0/0* by DIGTYPER and SVTyper, and not discovered by DELLY are *not* false negatives but, most likely, constitute variants simple absent in the studied individual.

In all cases, most of the variants were genotyped as *0/0* by both DIGTYPER and SVTyper. For the majority of those, no matching DELLY variant was found and therefore they are likely to be correctly genotyped as absent (gray area). In all cases, there is a sizeable overlap between data base variants discovered by DELLY and

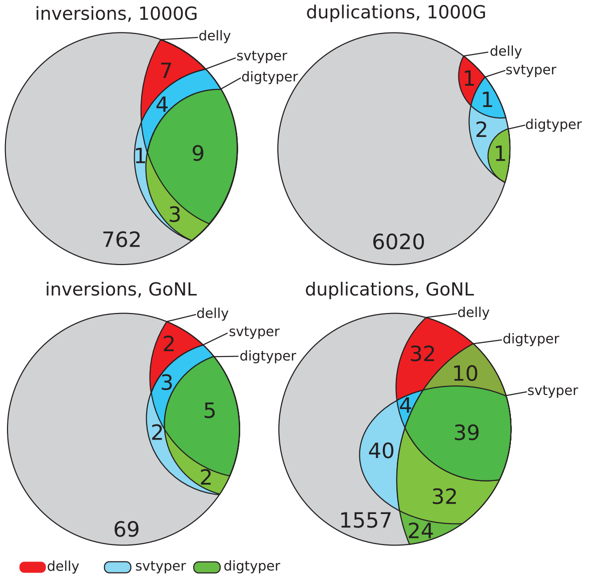


Fig. 8. Venn diagram of data base variants genotyped to be present in the GIAB individual by different methods. The different charts show inversions (left) and duplications (right) provided by The 1000 Genomes Project Consortium (2015) (top) and by The Genome of the Netherlands Consortium (2014) (bottom). The gray areas represent variants genotyped as *0/0* or*./.* by DIGTYPER and SVTYPER, and not found by DELLY. The regions colored in red, blue and green show variants for which DELLY (red), SVTyper (blue) and DIGTYPER (green) predicted the variant to be present (genotypes *1/1,1/0* or *1/.*) (Color version of this figure is available at *Bioinformatics* online.)

variants typed to be present by both DIGTYPER and SVTyper. Variants in this set are most likely truly present in this individual. Combined these two areas of putatively correct results represent large fractions of the total variants and indicate 89.2–99.9% agreement. As is not uncommon for Venn diagrams of variant prediction methods, there is also a sizeable difference of variants typed to be present by one or two of the methods, but not all. The true status of these variants remains unknown to us, but one might hypothesize that not all these variants are false positives and that the methods therefore complement each other.

3.2.1 Runtime

In order to use DIGTYPER with split read analysis, we re-mapped the all reads using LASER. This took about 225.1 single-core hours on an Intel Xeon CPU E5-2670v2 running Linux (kernel 4.4.34). SVTyper expects the reads to be mapped with BWA MEM (Li, 2013). Since in the GIAB dataset reads were aligned with novoalign, we needed to realign them using BWA MEM. Furthermore, we needed to add mate tags, for which we used samblaster (Faust and Hall, 2014). In total, this took 169.83 h single-core hours. The times needed for genotyping all inversions and duplications were 27.1 and 25.7 min for DIGTYPER and SVTyper, respectively. The runtime of DELLY was 13.5 h.

4 Conclusion

In this paper, we have presented a new method to genotype tandem duplications and inversions. The issue in this is that the short read data that provides evidence of the genotype is affected by uncertainties, which can decisively hamper the task. Here, we have addressed this by a sound statistical framework that aims to determine the correct genotype as the most likely one given the short read data. It is common to maximum likelihood estimation procedures that naive approaches have exponential runtime when taking data uncertainties into account. One important achievement of ours has been to provide a computation scheme that allows to determine the genotype in runtime linear in the supporting short read data. As results, we have demonstrated that our method achieves significant improvements over DELLY, Pindel and SVTyper, to date the only methods that allow to genotype tandem duplications and inversions, in various aspects.

Still, there is room for improvements. For example, inversions, duplications and deletions often come in combination, which we have not addressed here. Since our approach is flexible in terms of combining variants, we will be able to address also this case in the future. We consider it worthwhile to further invest in re-aligning reads so as to achieve refined alignment probabilities and even more accurate read alignments. Extending our read mapper LASER to also detect split alignments for duplications could potentially bring an improvement. Last but not least, marrying our duplication genotyping approach to coverage-based techniques is a promising future endeavor, for instance by using coverage signals to obtain priors.

Funding

A.S. acknowledges funding from the Dutch Scientific Organization (NWO) through Vidi grant No. 639.072.309.

Conflict of Interest: none declared.

References

Carlson,K.D. *et al.* (2015) MIPSTR: a method for multiplex genotyping of germline and somatic STR variation across many individuals. *Genome Res.*, **25**, 750–761.

Catchen,J.M. *et al.* (2011) Stacks: building and genotyping loci de novo from short-read sequences. *G3*, **1**, 171–182.

Chiang,C. *et al.* (2015) SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nat. Meth.*, **12**, 966–968.

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.

Faust,G.G. and Hall,I.M. (2014) SAMBLASTER: fast duplicate marking and structural variant read extraction. *Bioinformatics*, **30**, 2503–2505.

Garrison,E. and Marth,G. (2012). Haplotype-based variant detection from short-read sequencing. arXiv:1207.3907 [q-bio].

Handsaker,R. *et al.* (2011) Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nat. Genet.*, **43**, 269–276.

Hehir-Kwa,J.Y. *et al.* (2016) A high-quality human reference panel reveals the complexity and distribution of genomic structural variants. *Nat. Commun.*, **7**, 12989.

Iqbal,Z. *et al.* (2012) De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.*, **44**, 226–232.

Layer,R.M. *et al.* (2014) LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.*, **15**, R84.

Li,H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio]. arXiv: 1303.3997.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

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

Lin,K. *et al.* (2015) Making the difference: integrating structural variation detection tools. *Brief. Bioinformatics*, **16**, 852–864.

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

Marschall,T. *et al.* (2013) MATE-CLEVER: Mendelian-inheritance-aware discovery and genotyping of midsize and long indels. *Bioinformatics*, **29**, 3143–3150.

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

Nuttle,X. *et al.* (2013) Rapid and accurate large-scale genotyping of duplicated genes and discovery of interlocus gene conversions. *Nat. Meth.*, **10**, 903–909.

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

Rimmer,A. *et al.* (2014) Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat. Genet.*, **46**, 912–918.

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 (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74.

The Genome of the Netherlands Consortium (2014) Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nat. Genet.*, **46**, 818–825.

The UK10K Consortium (2015) The uk10k project identifies rare variants in health and disease. *Nature*, **526**, 82–90.

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.

Zook,J.M. *et al.* (2015). Extensive sequencing of seven human genomes to characterize benchmark reference materials. bioRxiv. doi:10.1101/026468.