DeepTrio: Variant Calling in Families Using Deep Learning

Alexey Kolesnikov[1], Sidharth Goel[1], Maria Nattestad[1], Taedong Yun[1], Gunjan Baid[1], Howard Yang[1], Cory Y McLean[1], Pi-Chuan Chang[1], Andrew Carroll[1]

1. Google LLC, 1600 Amphitheatre Pkwy, Mountain View, CA
Correspondence: awcarroll@google.com

Abstract

Every human inherits one copy of the genome from their mother and another from their father. Parental inheritance helps us understand the transmission of traits and genetic diseases, which often involve *de novo* variants and rare recessive alleles. Here we present DeepTrio, which learns to analyze child-mother-father trios from the joint sequence information, without explicit encoding of inheritance priors. DeepTrio learns how to weigh sequencing error, mapping error, and *de novo* rates and genome context directly from the sequence data. DeepTrio has higher accuracy on both Illumina and PacBio HiFi data when compared to DeepVariant. Improvements are especially pronounced at lower coverages (with 20x DeepTrio roughly equivalent to 30x DeepVariant). As DeepTrio learns directly from data, we also demonstrate extensions to exome calling solely by changing the training data. DeepTrio includes pre-trained models for Illumina WGS, Illumina exome, and PacBio HiFi.

Introduction

Genomic sequencing can identify variants informative[1] for diseases[2], traits[3], and ancestry[4]. Sequencing is particularly informative in rare genetic disease[5] caused by high-impact pathogenic variants[6]. In a mother-father-child trio, each parent contributes half of their genome, with the addition of a small number of *de novo* variants[7]. Sequencing for rare disease often includes the parents in order to use this information for accurate variant identification and interpretation[8]. Rare disease studies analyze multiple families with the same suspected disease to resolve undiagnosed cases[9].

A number of methods can discover germline variants in an individual sample. Traditional methods model the evidence for a variant with known contributors to uncertainty, such as the rate of sequencing errors[10], the probability that a read is mapped incorrectly[11], and the reliability of the sequence quality scores[12]. Software tools which employ these statistical approaches include Freebayes[13], GATK[14], Octopus[15], 16GT[16], and Strelka2[17].

Recent approaches have used deep learning, which learns representations directly from data[18]. This allows a variant caller to capture aspects of the problem which are incompletely understood, or to rapidly adapt to a new sequencing technology by training on data. Deep learning variant callers include DeepVariant[19], Clairvoyante[20], and Neusomatic[21]. Deep learning was used in a majority of short-read and virtually all long-read submissions to the PrecisionFDA Truth Challenge V2[22].

Some variant callers can use information about a trio to jointly call variants. The approaches range from joint calling without consideration of family information (e.g. GATK GenotypeGVCF), those which model parental transmission probabilities (e.g, FamSeq, GATK CalculateGenotypePosteriors), and those which use a deep learning approach for individual samples and postprocess with statistical methods (e.g. dv-trio[23])

Correctly incorporating parental information requires integrating the existing uncertainties in error sources across multiple samples. A deep learning approach can directly learn from trio data how to value evidence from parents in making a call. It is also easy to adapt to different coverages, preparations, and technologies.

In this work, we build and assess DeepTrio, a deep learning-based variant caller for parent-child trios. We start from the code base of DeepVariant, a germline caller which won multiple awards in the PrecisionFDA Truth Challenge V2[22], noted for high accuracy on genomes and exomes[24], and shown to increase detection rate of pathogenic germline variants[25].

We train DeepTrio to call variants in both parent and child samples, with one model for Illumina WGS, one for Illumina exome, and one for PacBio HiFi[26]. To ensure accurate performance over a range of conditions, DeepTrio is trained with a diversity of preparations (PCR-free, PCR-positive, and multiple exome kits), and across a range of child and parent coverages. DeepTrio is also trained for duo-calling. DeepTrio can write output as individual VCFs[27], gVCFs, or as a merged family VCF. The gVCFs of multiple families can be combined with GLnexus[28], which has been optimized for combining DeepVariant gVCFs[29], to scalably create large joint callsets of trios, duos, and individual samples.

We show that DeepTrio has superior accuracy to both individual sample and trio-based samples, measured by concordance with the Genome in a Bottle truth set[30,31]. We show that DeepTrio is still able to accurately call *de novo* variants, despite their lack of support in the parents. Finally, we quantify the performance of DeepTrio across coverage, showing that DeepTrio allows high accuracy to be retained at lower coverage for both proband and parent samples.

Results

Modifying DeepVariant to call trios

DeepVariant calls variants in three steps: **make_examples**, **call_variants**, and **postprocess_variants**. In the **make_examples** step, a simple heuristic identifies positions which might differ from the reference. For Illumina sequencing, this requires a fraction of reads supporting an alternate allele at 0.12 for SNPs and 0.06 for Indels. For PacBio sequencing, the threshold is 0.12 for both SNPs and Indels.

The **call_variants** step represents BAM data as a multi-dimensional pileup of a 221-bp window in the genome around a candidate. As of DeepVariant v1.0, there are 8 input channels representing 1) the bases in the read, 2) their base quality, 3) the mapping quality of the read, 4) the strand mapped to, 5) whether the read supports a variant, 6) bases which differ from the reference and (in the case of PacBio data) 7,8), realignments of the reads to the alternate alleles. **postprocess_variants** converts the output probabilities of the neural network into a variant call and confidence, and resolves multi-allelic candidates into their most likely alleles.

To modify **make_examples** for Trio calling, we perform candidate generation on each individual sample in the same manner. We also generate candidates from the union of reads from all samples with a reduced threshold for reads supporting the alternate allele, which allows discovery of alleles at a lower fraction but which are reinforced by appearing in multiple samples. When a candidate allele is identified in any single sample, it is also generated at the same position in the other samples. This allows us to generate an output variant probability for every candidate in every sample.

The **call_variants** stage uses a deep neural network to classify the probabilities for the genotype of each variant candidate. This process learns the important factors for classification directly from the input data. We generate tensor pileups where each sample has a fixed height with the child pileup in the middle, one parent's pileup on top, and the other parent at the bottom. Using labels from Genome in a Bottle, we train two models: a child model and a parent model. To generate calls for each parent, two sets of examples are made, with a different parent as the top pileup.

The concept of Mendelian inheritance, or any explicit modeling of parent-child relationship is never provided to DeepTrio. Training simply creates these pileups and associated labels. The child model would learn that the reads in the middle pileup are most informative for calling, and the parent reads as supporting evidence. The parent model would learn that the topmost reads are most informative for the call, with child reads providing some supporting information, and the other parent marginal additional information.

The **postprocess_variants** stage is not altered. The final outputs are a VCF and a gVCF for each sample. Merging multiple gVCFs uses GLnexus[28] in a manner optimized for DeepVariant outputs[29]. Figure 1 shows a representation of the calling process.

DeepTrio is trained on Genome in a Bottle samples[32] with sequencing conducted on Novaseq, HiSeqX, HiSeq4000, and PacBio Sequel II instruments with both PCR-Free and PCR+ preparations. Exome models are trained on Agilent SureSelect v7, IDT-xGen, and Truseq capture kits. This data has been previously described and released[33]. DeepTrio is trained on examples from chromosomes 1-19. Chromosomes 21 and 22 are used as a tuning set to select the model checkpoint by determining when accuracy on the withheld set has peaked. Chromosome 20 is fully withheld as an independent evaluation set to assess accuracy.

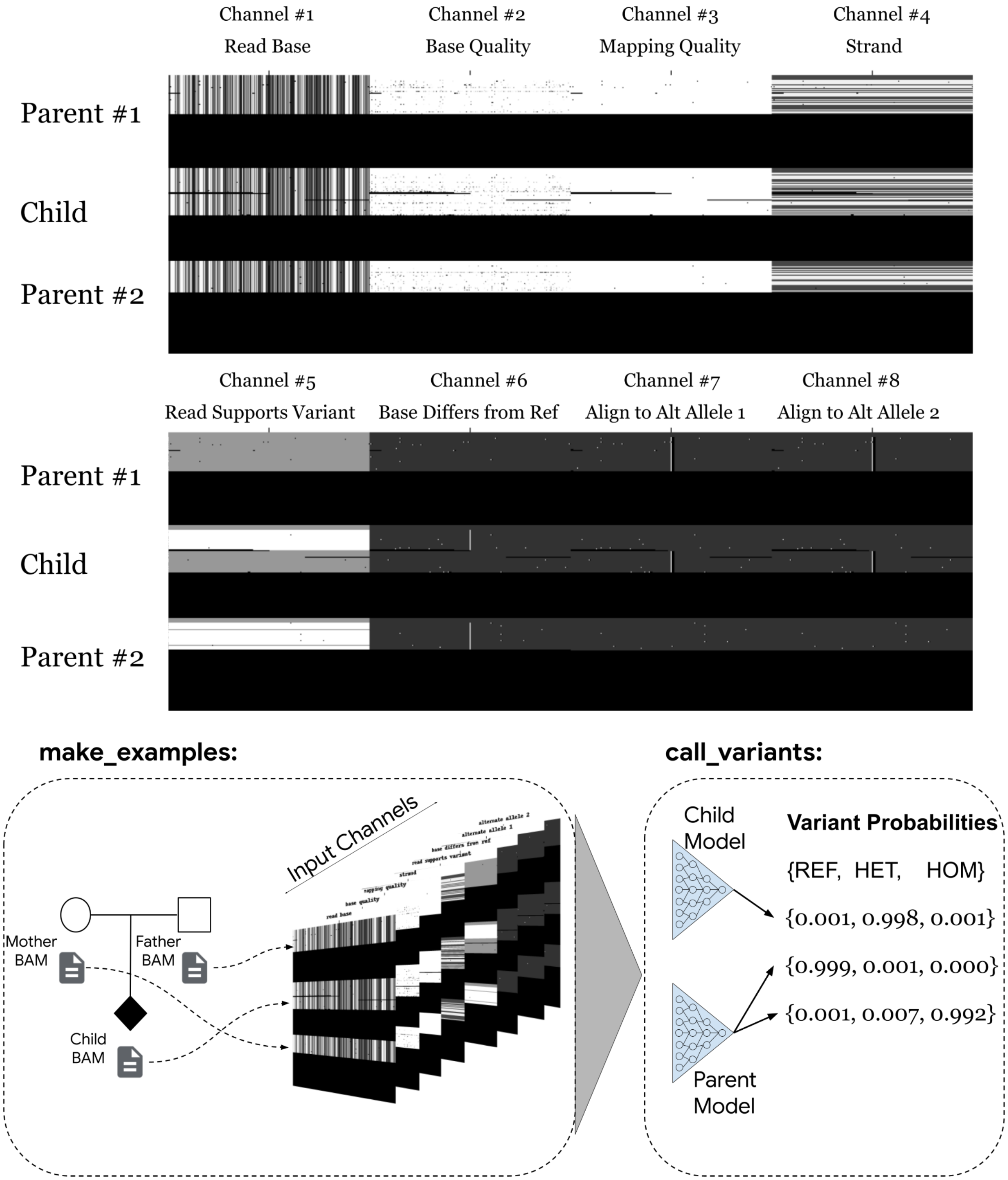


Figure 1. DeepTrio inputs channels and processing pipeline
DeepTrio represents data from the BAM file of a child and one or two parents as a pileup of a 221-bp long window (x-axis) with reads (y-axis). Each sample has a fixed height with parent and child reads in a different row. DeepTrio presents 8 input channels (z-axis). Example shown is PacBio HiFi (top). One model is trained to call variants in the child and another in the parent. DeepTrio makes multiple examples, alternating the position of each parent (bottom).

Assessing variant calling accuracy

To determine the improvements of trio calling, we compared DeepTrio to DeepVariant, GATK4 HaplotypeCaller (non-trio), GATK4 CalculateGenotypePosteriors (trio), dv-trio, Octopus, and FamSeq for the Genome in a Bottle (GIAB) Ashkenazi Jewish trio (HG002-HG003-HG004) datasets from the PrecisionFDA v2 Truth Challenge[22]. Accuracy is determined by concordance with the GIAB v4.2.1 truth set[30,31,34] using hap.py[35].

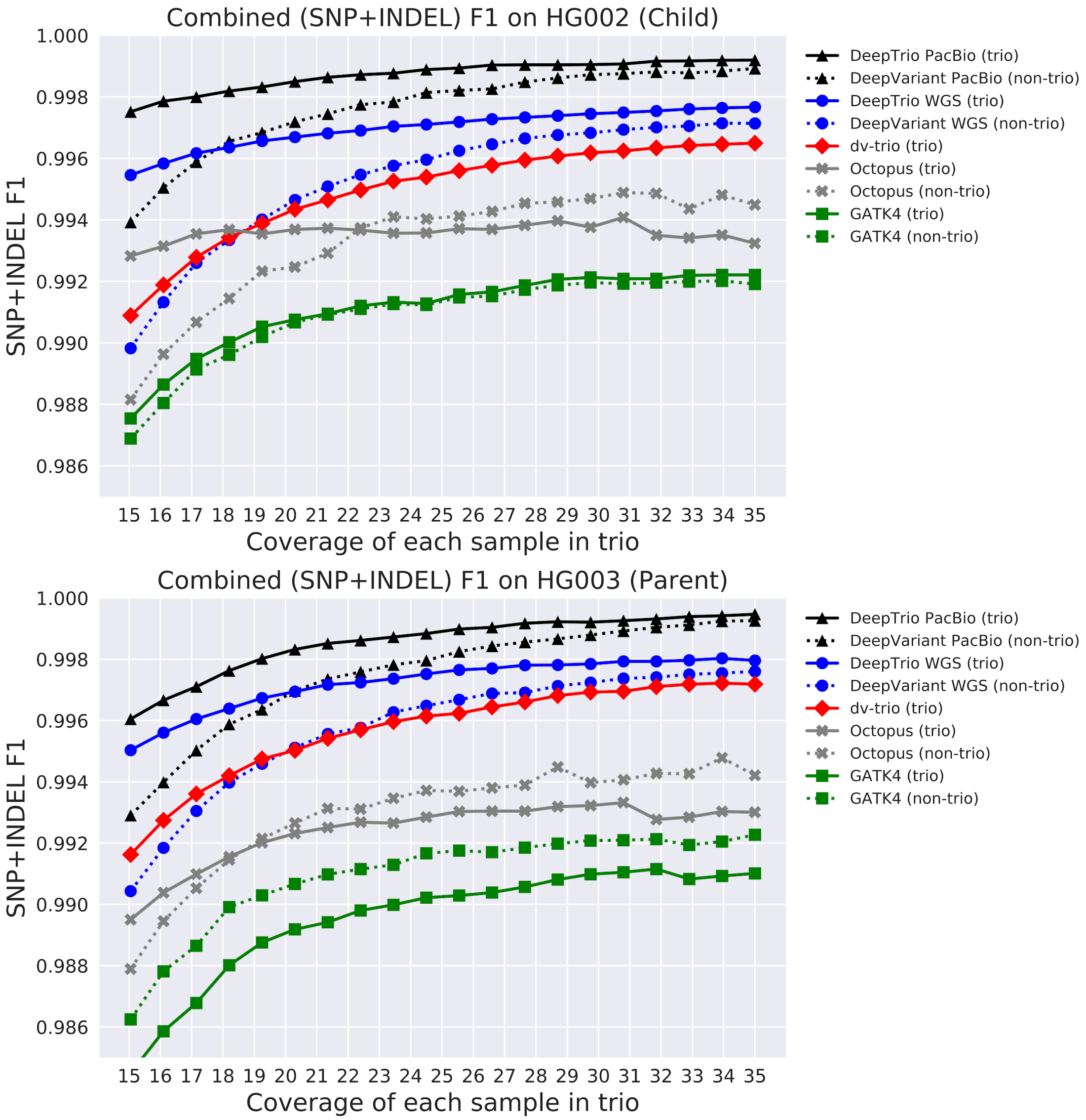


Figure 2. Variant calling accuracy of DeepTrio varying depth of all trio samples
Accuracy for DeepTrio and other pipelines over coverages, determined by concordance with the Genome in a Bottle v4.2 truth set for the child (top) and parent (bottom), with the same coverage on all three samples. All samples use the same Illumina data, except for the black lines DeepVariant PacBio and DeepTrio PacBio. Trio methods are shown with a solid line and non-trio with a dotted line. F1 is determined from the total errors and correct calls for total Indels and SNPs for chromosome20. The F1 for FamSeq was much lower, and was excluded.

Table 1. HG002 SNP accuracy at 35 x Coverage (all trio members)

Tool	Data Type	F1	Recall	Prec	FN	FP	FP. gt	TP de novo	FP de novo
DeepTrio (trio)	Illumina	0.9979	0.9962	0.9996	269	29	4	31/34	3
DeepVariant	Illumina	0.9973	0.9953	0.9993	334	48	12	34/34	20
GATK4 (trio)	Illumina	0.9921	0.9947	0.9895	380	754	61	32/34	11
GATK4	Illumina	0.9917	0.9943	0.9892	407	775	62	34/34	144
dv-trio (trio)	Illumina	0.9964	0.9946	0.9983	385	122	38	0/34	0
Octopus (trio)	Illumina	0.9935	0.9948	0.9923	370	551	33	21/34	0
Octopus	Illumina	0.9943	0.9956	0.9929	309	494	31	-	-
Famseq (trio)	Illumina	0.9653	0.9857	0.9458	1021	4033	622	31/34	10
DeepTrio-PB	PacBio	0.9994	0.9992	0.9997	60	20	16	34/34	0
DeepVariant-PB	PacBio	0.9995	0.9995	0.9998	60	15	9	34/34	0

Table 2. HG002 Indel accuracy at 35 x Coverage (all trio members)

Tool	Data Type	F1	Recall	Prec	FN	FP	FP. gt	TP de novo	FP de novo
DeepTrio (trio)	Illumina	0.9971	0.9957	0.9985	48	18	11	3/4	1
DeepVariant	Illumina	0.9964	0.9946	0.9982	61	21	15	4/4	2
GATK4 (trio)	Illumina	0.9926	0.9931	0.9922	78	92	36	3/4	1
GATK4	Illumina	0.9929	0.9926	0.9933	83	79	34	4/4	11
dv-trio (trio)	Illumina	0.9955	0.9932	0.9979	77	24	15	4/4	2
Octopus (trio)	Illumina	0.9941	0.9947	0.9936	60	74	12	1/4	0
Octopus	Illumina	0.9939	0.9931	0.9946	73	60	24	-	-
Famseq (trio)	Illumina	-	-	-	-	-	-	-	-
DeepTrio-PB	PacBio	0.9950	0.9963	0.9938	42	73	24	3/4	2
DeepVariant-PB	PacBio	0.9913	0.9913	0.9916	95	93	57	2/4	1

We observe that DeepTrio has higher accuracy than DeepVariant for both Illumina and PacBio HiFi data (Figure 2 top, Tables 1 and 2), with a more pronounced effect at lower coverage depths (Supplementary Tables 1 and 2).

Assessing variant calling accuracy in the parent samples

Accurate variant calling of parent samples is important in order to give context for proband variants, to accurately catalog incidental findings, and to correctly generate research cohorts. To assess the accuracy of parent calling, we compared DeepTrio's parent model performance with DeepVariant and other trio and non-trio methods (Figure 2 bottom). DeepTrio outperforms other pipelines across a range of coverage, with a larger effect at lower coverages. DeepTrio's advantage is less pronounced for the parent model as compared to its advantage on the child model. This finding is reasonable, since the genotype of the child is less informative regarding the genotype of the parent than the parent's genotype is informative regarding the genotype of the child. The improvement in accuracy allows a lower coverage for the parent sample to be used while retaining the accuracy one would normally achieve with a non-trio pipeline.

Precision and recall of *de novo* variants

The ability to identify *de novo* variants which may have a dominant inheritance pattern is of particular interest in identifying rare genetic disease. Because *de novo* variants violate the assumptions of Mendelian inheritance, it is reasonable to think that trio calling approaches may have a more pronounced effect on the sensitivity and specificity of *de novo* variants. When the analysis is constrained to cases where a child is called as 0/1 and each parent is confidently called as 0/0, we observe far more pronounced differences between trio and non-trio pipelines, even in cases like GATK4 where the overall accuracy as determined by Genome in a Bottle comparison is very similar.

The trio-aware pipelines (DeepTrio, dv-trio, famseq, GATK CalculateGenotypePosteriors), and Octopus have a greatly reduced false positive rate for *de novo* variants, but have slightly reduced recall of true *de novo* variants, whether these are defined as identifying the call as *de novo* by confidently genotyping the child as 0/1 and the parents as 0/0, or by detecting the child variant, but with an unknown or incorrect all in a parent (Figure 3, Tables 1,2, Supplementary tables 1,2).

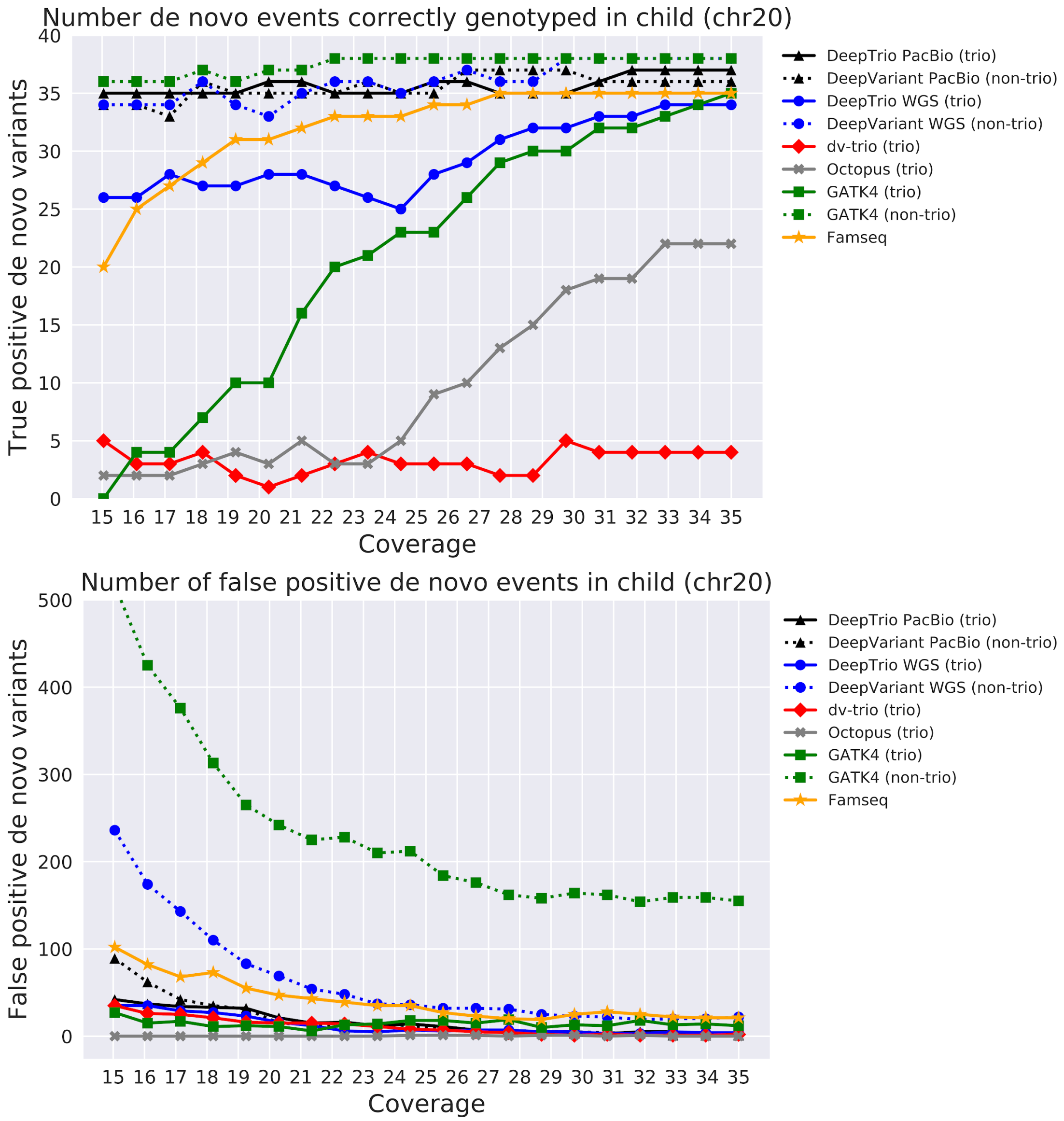


Figure 3. True positive and false positives for *de novo* variants
The number of false positive de novo variants (confident genotype calls of child 0/1, parents 0/0) called in each pipeline (top left). The number of de novo variants in Genome in a Bottle with confident genotype calls for each individual (child 0/1, parents 0/0) (top right).

Assessing the effect of parental depth on variant calling accuracy in the proband

In trio sequencing for rare disease, there is often greater importance in sequencing a child proband. To manage sequencing costs, studies often take the approach of sequencing the parents at a lower coverage than the child[36]. In order to evaluate performance in these scenarios, we performed downsampling of the parent samples while keeping the child coverage at 35x. Variant calls were generated with the same tools used for the coverage titration of the full trio and evaluated using the same methods.

For the child sample, which contains the same reads for each titration point, we observe a slight accuracy improvement for DeepTrio in both Illumina and PacBio HiFi across the parent coverage ranges observed (15x-35x). For the parent samples, which do vary in coverage over the titration range, we observe a much greater advantage for DeepTrio compared to DeepVariant and other methods (Figure 4). For PacBio HiFi, the parent model has higher accuracy when the child is at 35x, as compared to when all three samples are coverage-titrated.

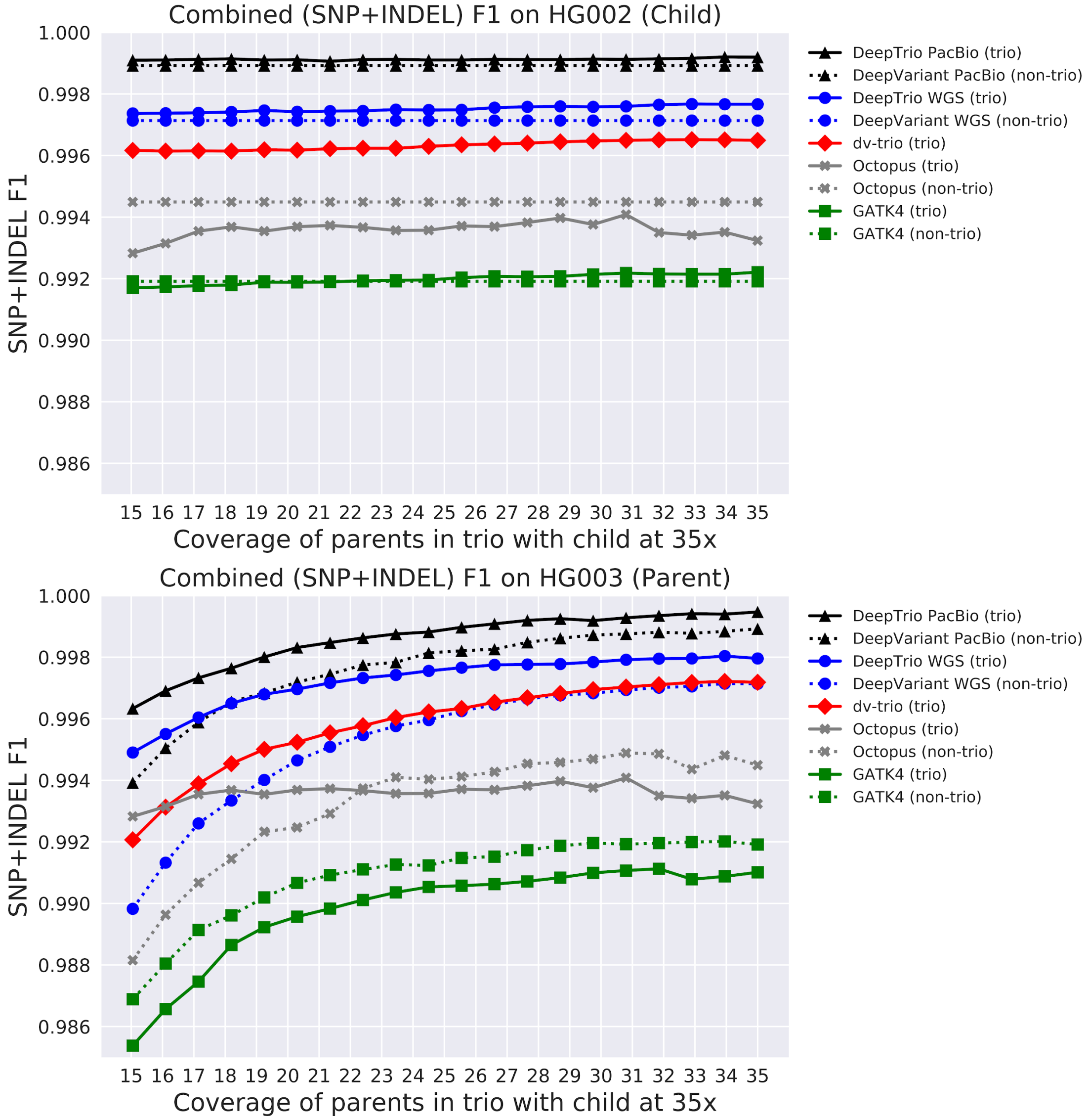


Figure 4. Variant calling accuracy of DeepTrio varying only parent coverage
Accuracy for DeepTrio and other pipelines over coverage titrations of the parent samples with the child sample at 35x. All samples use the same Illumina data, except for the black lines DeepVariant PacBio and DeepTrio PacBio. Accuracy is determined by concordance with the Genome in a Bottle v4.2 truth set for the child (bottom left) and parent (bottom right). Trio methods are shown with a solid line and non-trio with a dotted line. F1 is determined from the total errors and correct calls for Indels added to SNPs for chromosome20. The F1 for FamSeq was much lower and was excluded.

Inspecting examples of variant calls improved by DeepTrio

In order to better understand cases where DeepTrio makes a correct call where other methods do not, we inspected IGV[37] images of positions which DeepTrio called correctly that were errors in DeepVariant. Since the F1 of SNP calling at 35x is already 0.9973, all inspected sites were difficult (low coverage, low mappability, presence of repeats and segmental duplications). Errors corrected by DeepTrio were often either through the ability to identify supporting evidence at low coverage and difficult to map regions (Figure 5), or the ability to better estimate the correct genotype in sites with substantial allelic bias (Supplementary Figure 1).

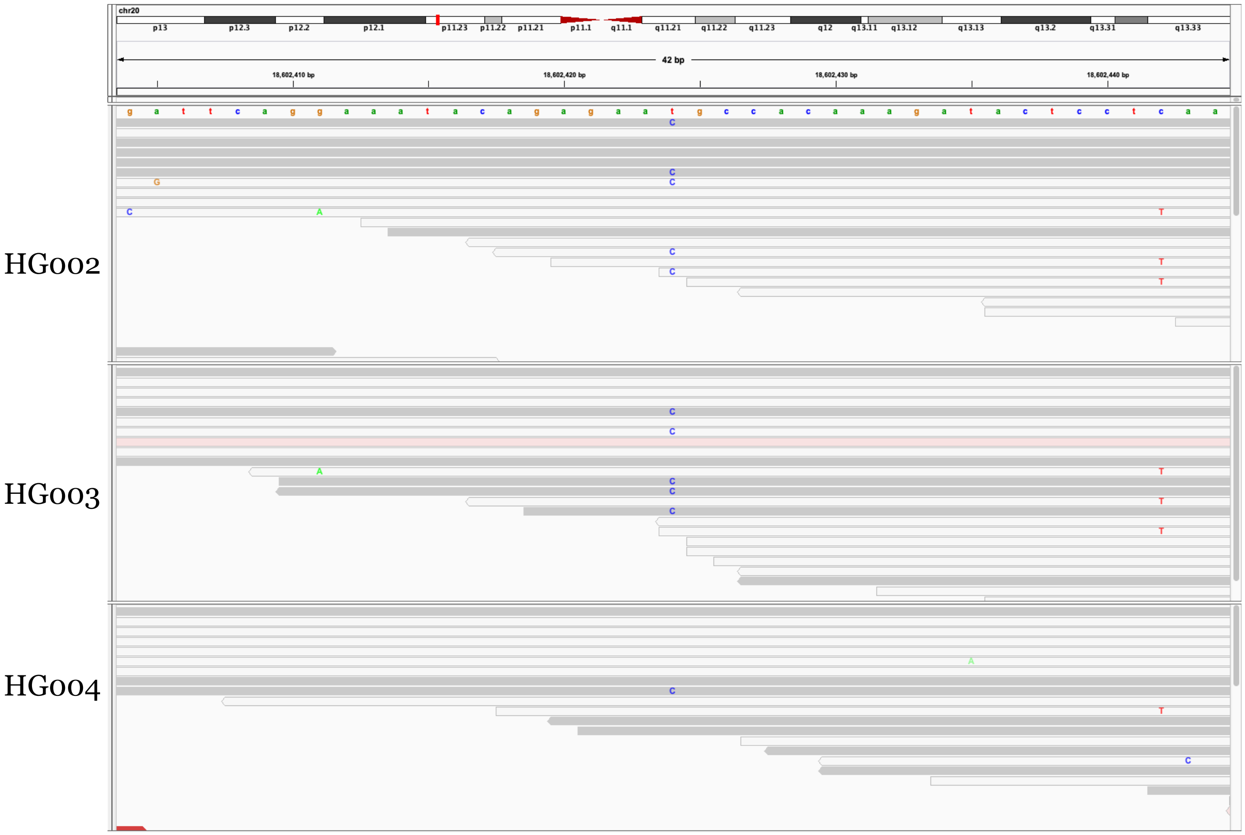


Figure 5. Example of a variant called correctly by DeepTrio but not DeepVariant
IGV image of chr20:18602424 in 35x Illumina WGS PrecisionFDA v2 Truth Challenge samples for HG002-HG003-HG004. HG002 (child) is shown in the top row. This position is not called as a variant in DeepVariant and is correctly called as a heterozygous variant in DeepTrio. IGV marks reads with MAPQ 0 as white instead of gray, indicating that this region is difficult to map with Illumina reads.

Computational efficiency of DeepTrio and other trio-calling pipelines

To assess the computational efficiency of DeepTrio compared to other methods of generating trio calls, we ran pipelines on the Illumina 35x WGS samples using the same hardware: a 16-CPU thread machine (n1-standard-16) available on Google Cloud Platform. This was chosen as representative of typical runs, though there are faster (and more expensive) or slower (but cheaper) methods to run these pipelines (an analysis of single-machine scaling can be found in the DeepVariant-GLnexus paper[38]).

Some components of DeepTrio are more computationally efficient when compared to DeepVariant, for example DeepTrio can reuse calculations in the example generation stage. Other components require more compute in DeepTrio: the size of the pileup images is larger, which corresponds to a larger neural network and more compute. Also, more examples are identified across the samples and any example in one sample requires making a call in the other. We observe that DeepTrio requires more time to run when compared to DeepVariant, but slightly less time than GATK4 when using these hardware settings (Figure 6).

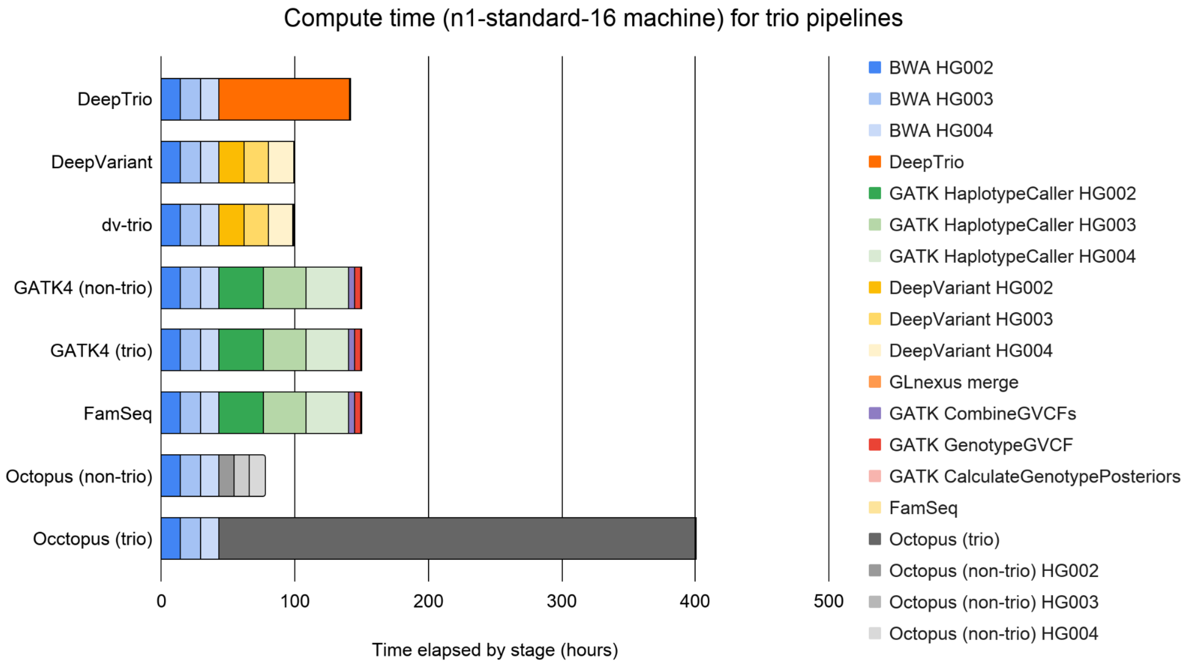


Figure 6. Time required to run DeepTrio and other pipelines
Compute time required for each stage of trio calling pipelines for a 35x Illumina WGS trio of HG002-HG003-HG004. Analysis is conducted on the full genome. The same machine type is used in each case, a 16-CPU thread instance n1-standard-16.

Accuracy on ChromosomeX

The Genome in a Bottle truth sets do not contain chromosomeX or chromosomeY variants in a male individual. As a result, DeepTrio has never been trained with hemizygous sites. Because we train DeepTrio to perform duo calling, it is likely that DeepTrio would call variants on chromosomeX similar to how it would call a duo sample. To assess this, we ran DeepVariant and DeepTrio on chromosomeX of the son (HG002) and measured the number of heterozygous variant calls in the non-PAR regions of chromosomeX.

For DeepVariant, 4.45% (455/101866) of calls in non-PAR regions of chromosomeX are heterozygous. In DeepTrio, 24.5% (21633/88314) are heterozygous. This substantial difference suggests that applying DeepTrio directly to chromosomeX in male samples is problematic.

Since chromosomeX in males is inherited from the mother, we performed calling on chromosomeX with only the mother provided as the parent. This reduced heterozygous calls to 3.37% (3518/104427), which is better than in the DeepVariant case. For male samples, this recommends that variant calling should be run with both parents on the autosomal and PAR regions using a BED file to restrict location, and additional variant calling should be performed using only the mother's file provided as parent for the non-PAR regions of chromosomeX, and only the father's provided for the non-PAR regions of chromosomeY.

This experiment indicates that allowing the model to infer a hemizygous chromosome through coverage and explicitly training for hemizygous variants is an opportunity for improvement, both for DeepVariant and DeepTrio.

Discussion

Here we discuss how DeepTrio's performance characteristics relate to the motives for trio calling and the data which is generally available. In this work, we considered accuracy across all variants, and separately for *de novo* variants. In the context of rare disease, both of these formulations of accuracy are important. Some rare genetic diseases are caused by the combination of recessive variants from each parent. While others, especially those with a dominant inheritance pattern, will arise *de novo*.

We demonstrate that DeepTrio has strictly superior accuracy compared to DeepVariant and other trio and non-trio methods. When considering highest overall accuracy, DeepTrio would always be preferred. The case of *de novo* variants is more nuanced. DeepTrio has much higher precision when calling *de novo* variants but has slightly lower recall. If only F1 is considered, DeepTrio's F1 (0.8947) is superior to the next highest trio method (GATK4 CalculateGenotypePosteriors - 0.8235), and non-trio method (DeepVariant - 0.7755). But in terms of recall, non-trio methods like DeepVariant have a higher recall of *de novo* calls. As a result, if investigators have a greater interest in the highest recall of *de novo* events, as opposed to overall accuracy, one option would be to run DeepVariant on each sample, and to re-run DeepTrio on the smaller number of regions where DeepVariant identified a *de novo* variant, in order to prioritize real variants first. In addition, we have previously shown that the output of the neural network is very well calibrated (Figure 2 of *poplin et al. 2018*)[19], and it may be possible to rank putative *de novo* events by the reported confidence of the call in order to tune DeepTrio more towards recall of *de novo*.

The accuracy analyses above are conducted at deep coverage (35x). Sequencing a trio of samples is more expensive than a single sample, and studies often compensate for this extra expense by sequencing at a lower coverage, especially for the parents. Reducing coverage is often considered when sequencing non-human disease samples, where the importance of accurately calling every variant is balanced against larger, more comprehensive studies. DeepTrio's advantage over other methods is substantially greater at lower coverages. DeepTrio is about as accurate in ~20x coverage as DeepVariant is at 28-30x coverage for both Illumina WGS and PacBio HiFi. DeepTrio is more accurate at 15x coverage than either GATK4 method at the highest coverage evaluated (35x). By maintaining high accuracy at reduced coverage, as well as by including training examples which reduce parent coverage while keeping child coverage, DeepTrio increases the flexibility of investigators to plan their trio sequencing to maximize cost-benefit.

As a deep learning method, DeepTrio does not explicitly encode the relationship between samples. DeepTrio's ability to improve accuracy of calling, and to do so in a manner which is similar to human intuition regarding *de novo* variants, demonstrates an ability to capture rules which mirror general knowledge. This is similar to a recent demonstration which re-trained DeepVariant to use population allele frequencies[39]. It is a strong indicator that deep-learning based variant callers can be further improved by finding ways to expose information which captures the underlying biology of samples and populations. Similarly, the framework of DeepTrio could, in theory, be further expanded to use sibling information, or to leverage more distant family relationships. Overall, the success of DeepTrio is a strong demonstration that thoughtfully identifying data which captures relevant biological or bioinformatics intuition, is a critical element to the development of strong machine learning methods in the genomics domain.

Methods

Generation of Sequencing Data

The generation of sequencing data for training is described in detail in *Baid et al. 2020* [33] and the WGS and PacBio evaluation data in *Olson et al. 2020*[22]. In summary, all WGS and exome runs were conducted with 151-bp paired-end reads at 50x intended coverage from NovaSeq and HiSeqX platforms. For WGS, sequencing for both PCR-Free and PCR-Positive preparations. All sequencing was performed on HG001-HG007, NA12891, and NA12892. For exomes, sequencing was performed with multiple capture kits, Agilent v7, IDT-xGen, and Nextera at a target of 200x coverage from NovaSeq and HiSeq4000 platforms.

For PacBio HiFi data, we requested 3 SMRT Cells 8M for each sample of HG003, HG004, HG006, and HG007. Libraries were prepared targeting a 15kb insert size and sequenced on

Sequel II System with Chemistry 2.0. This was supplemented by data from Human Pangenome Reference Consortium (https://github.com/human-pangenomics/HG002_Data_Freeze_v1.0) .

Training models

Training DeepTrio requires a set of BAM or CRAM files and a set of truth labels. Examples are generated in the same manner used for calling and are annotated with the truth labels. A training tutorial for DeepVariant from input data is available at: (https://github.com/google/deepvariant/blob/r1.1/docs/deepvariant-training-case-study.md).

Using the trio data sets described in the prior section, DeepTrio was trained across a range of coverages achieved by random downsampling of ~50x BAM files at fractions of 0.7 (~35x), 0.5 (~25x), and 0.3 (~15x). This random downsampling helps DeepTrio to generalize well across coverages, and we observe that having more difficult examples results in overall better models. Examples are generated for trios where each sample is downsampled at the same fraction, and those where only the parents are further downsampled while the child is kept at a higher coverage. To train DeepTrio to natively handle duo calling, training also occurs in the same manner, but omitting one of the parents. Training occurs over the entire set of WGS samples to generate the WGS model.

For exomes, the data described in the prior section is used in the same manner as in WGS, but with different downsample fractions. The downsample fractions for exomes are 0.8, 0.6, and 0.4. The resulting coverages are diverse, since exome capture varies substantially by exon and sample. For PacBio, different combinations of 8M SMRT cell runs provide the training inputs, over a range of 2, 3, 4, 5, and 6 merged cells as the input files.

Mapping and Variant Calling

Samples were mapped to GRCh38[40] with BWA MEM[41] in an ALT-aware manner and deduplicated with Picard MarkDuplicates [42].

Variant calling was performed using DeepVariant v1.0[19], DeepTrio, GATK4.1.6.0[14], FamSeq[1] (https://github.com/wwylab/FamSeq/commit/63be74f39183077c98fceed97d71bf51dfb80929), and dv-trio v1.0.0[23]. Timing estimates for DeepVariant used DeepVariant v1.1, using the OpenVINO acceleration by Intel, a recent contribution which speeds execution.

For DeepVariant, calling was performed following DeepVariant's best practices in multi-sample calling (https://github.com/google/deepvariant/blob/r1.1/docs/trio-merge-case-study.md).

For GATK, non-trio aware calling was performed by HaplotypeCaller followed by GenotypeGVCFs. For GATK trio-aware calling, this VCF was further refined by CalculateGenotypePosteriors.

For FamSeq, GATK HaplotypeCaller and GATK GenotypeGVCFs were run, and FamSeq was run on the resulting multi-sample VCF. Since FamSeq only refines the call of SNPs, Indel calls were taken from the GATK VCF without modification.

For dv-trio, single sample calling was performed as described in DeepVariant's best practices in multi-sample calling, and the dv-trio scripts were applied to the output of this file.

For Octopus, single sample variant calling for single samples was performed using the v0.7.2 release, with the matched v0.7.2 germline forest model. Because Octopus does not generate a gVCF, we did not attempt to generate a joint call for the individual samples, to avoid introducing issues with harmonizing allele representation.

For Octopus trio calling, we ran both v0.7.2 and v0.7.0 and observed higher accuracy with v0.7.0. Benchmark numbers for Octopus are from the better performance we observed in v0.7.0.

For all *de novo* analyses, only PASS entries were used for calculations across all callers. No call (./.) positions were excluded.

Assessing accuracy

Call sets were assessed using v0.3.9 of the haplotype comparison tool, hap.py[35]. The v4.2.1 truth sets from GIAB[30,35] were used to benchmark HG002-4 samples mapped to GRCh38.

Code and Data Availability

All DeepTrio code is available under a BSD-3 license at:
https://github.com/google/deepvariant/tree/r1.1/deeptrio

All evaluation data are derived the Illumina and PacBio FASTQ files available from the PrecisionFDA v2 Truth Challenge[22] at: https://precision.fda.gov/challenges/10

Training datasets are described in "*An Extensive Sequence Dataset of Gold-Standard Samples for Benchmarking and Development*"[33] and download links for all sequence data are available in the supplement of that paper:
https://www.biorxiv.org/content/10.1101/2020.12.11.422022v1.supplementary-material

Acknowledgements

We thank Daniel Cook from Google for many helpful suggestions on the manuscript text and content. We thank Aaron Wenger and Billy Rowell for several helpful comments, especially suggestions to evaluate DeepTrio performance on chromosomeX.

Author contributions

AC, PC, and AK conceived and designed the study. AK, SG, MN, GB, and PC wrote DeepTrio code. TY and CYM built and analyzed variant merging methods. AK, PC, and AC analyzed results. AC and HY procured sequence data. AC wrote the manuscript.

Competing interests

All authors are employees of Google LLC and own Alphabet stock as part of the standard compensation package. This study was funded by Google LLC.

References

1. Peng, G., Fan, Y. & Wang, W. FamSeq: a variant calling program for family-based sequencing data using graphics processing units. *PLoS Comput. Biol.* **10**, e1003880 (2014).
2. Yang, Y. *et al.* Clinical whole-exome sequencing for the diagnosis of mendelian disorders. *N. Engl. J. Med.* **369**, 1502–1511 (2013).
3. Welter, D. *et al.* The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Res.* **42**, D1001–6 (2014).
4. Consortium, T. 1000 G. P. & The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature* vol. 467 1061–1073 (2010).
5. Boycott, K. M., Vanstone, M. R., Bulman, D. E. & MacKenzie, A. E. Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nat. Rev. Genet.* **14**, 681–691 (2013).
6. MacArthur, D. G. *et al.* Guidelines for investigating causality of sequence variants in human disease. *Nature* **508**, 469–476 (2014).
7. Acuna-Hidalgo, R., Veltman, J. A. & Hoischen, A. New insights into the generation and role of de novo mutations in health and disease. *Genome Biol.* **17**, 241 (2016).
8. Roach, J. C. *et al.* Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science* **328**, 636–639 (2010).

9. Gahl, W. A. *et al.* The National Institutes of Health Undiagnosed Diseases Program: insights into rare diseases. *Genet. Med.* **14**, 51–59 (2012).
10. Ma, X. *et al.* Analysis of error profiles in deep next-generation sequencing data. *Genome Biol.* **20**, 50 (2019).
11. Li, H. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics* **30**, 2843–2851 (2014).
12. Li, M., Nordborg, M. & Li, L. M. Adjust quality scores from alignment and improve sequencing accuracy. *Nucleic Acids Res.* **32**, 5183–5191 (2004).
13. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. *arXiv [q-bio.GN]* (2012).
14. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491–498 (2011).
15. Cooke, D. P., Wedge, D. C. & Lunter, G. A unified haplotype-based method for accurate and comprehensive variant calling. *Nat. Biotechnol.* (2021) doi:10.1038/s41587-021-00861-3.
16. Luo, R., Schatz, M. C. & Salzberg, S. L. 16GT: a fast and sensitive variant caller using a 16-genotype probabilistic model. *Gigascience* **6**, 1–4 (2017).
17. Kim, S. *et al.* Strelka2: Fast and accurate variant calling for clinical sequencing applications. doi:10.1101/192872.
18. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
19. Poplin, R. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983–987 (2018).
20. Luo, R., Sedlazeck, F. J., Lam, T.-W. & Schatz, M. C. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat. Commun.* **10**, 998 (2019).
21. Sahraeian, S. M. E. *et al.* Deep convolutional neural networks for accurate somatic mutation detection. *Nat. Commun.* **10**, 1041 (2019).
22. Olson, N. D. *et al.* precisionFDA Truth Challenge V2: Calling variants from short- and

long-reads in difficult-to-map regions. *Cold Spring Harbor Laboratory* 2020.11.13.380741 (2020) doi:10.1101/2020.11.13.380741.

23. Ip, E. K. K., Hadinata, C., Ho, J. W. K. & Giannoulatou, E. dv-trio: a family-based variant calling pipeline using DeepVariant. *Bioinformatics* **36**, 3549–3551 (2020).
24. Pedersen, B. S. *et al.* Effective variant filtering and expected candidate variant yield in studies of rare human disease. 2020.08.13.249532 (2020) doi:10.1101/2020.08.13.249532.
25. AlDubayan, S. H. *et al.* Detection of Pathogenic Variants With Germline Genetic Testing Using Deep Learning vs Standard Methods in Patients With Prostate Cancer and Melanoma. *JAMA* **324**, 1957–1969 (2020).
26. Wenger, A. M. *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat. Biotechnol.* **37**, 1155–1162 (2019).
27. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).
28. Lin, M. F. *et al.* GLnexus: joint variant calling for large cohort sequencing. 343970 (2018) doi:10.1101/343970.
29. Yun, T. *et al.* Accurate, scalable cohort variant calls using DeepVariant and GLnexus. *Cold Spring Harbor Laboratory* 2020.02.10.942086 (2020) doi:10.1101/2020.02.10.942086.
30. Wagner, J. *et al.* Benchmarking challenging small variants with linked and long reads. 2020.07.24.212712 (2020) doi:10.1101/2020.07.24.212712.
31. Zook, J. M. *et al.* An open resource for accurately benchmarking small variant and reference calls. *Nat. Biotechnol.* **37**, 561–566 (2019).
32. Zook, J. M. *et al.* Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data* vol. 3 160025 (2016).
33. Baid, G. *et al.* An Extensive Sequence Dataset of Gold-Standard Samples for Benchmarking and Development. *Cold Spring Harbor Laboratory* 2020.12.11.422022 (2020) doi:10.1101/2020.12.11.422022.

34. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246–251 (2014).

35. Krusche, P. *et al.* Best practices for benchmarking germline small-variant calls in human genomes. *Nat. Biotechnol.* **37**, 555–560 (2019).

36. Hiatt, S. M. *et al.* Long-read genome sequencing for the diagnosis of neurodevelopmental disorders. *Cold Spring Harbor Laboratory* 2020.07.02.185447 (2020) doi:10.1101/2020.07.02.185447.

37. Thorvaldsdóttir, H., Robinson, J. T. & Mesirov, J. P. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform.* **14**, 178–192 (2013).

38. Yun, T. *et al.* Accurate, scalable cohort variant calls using DeepVariant and GLnexus. *Bioinformatics* (2021) doi:10.1093/bioinformatics/btaa1081.

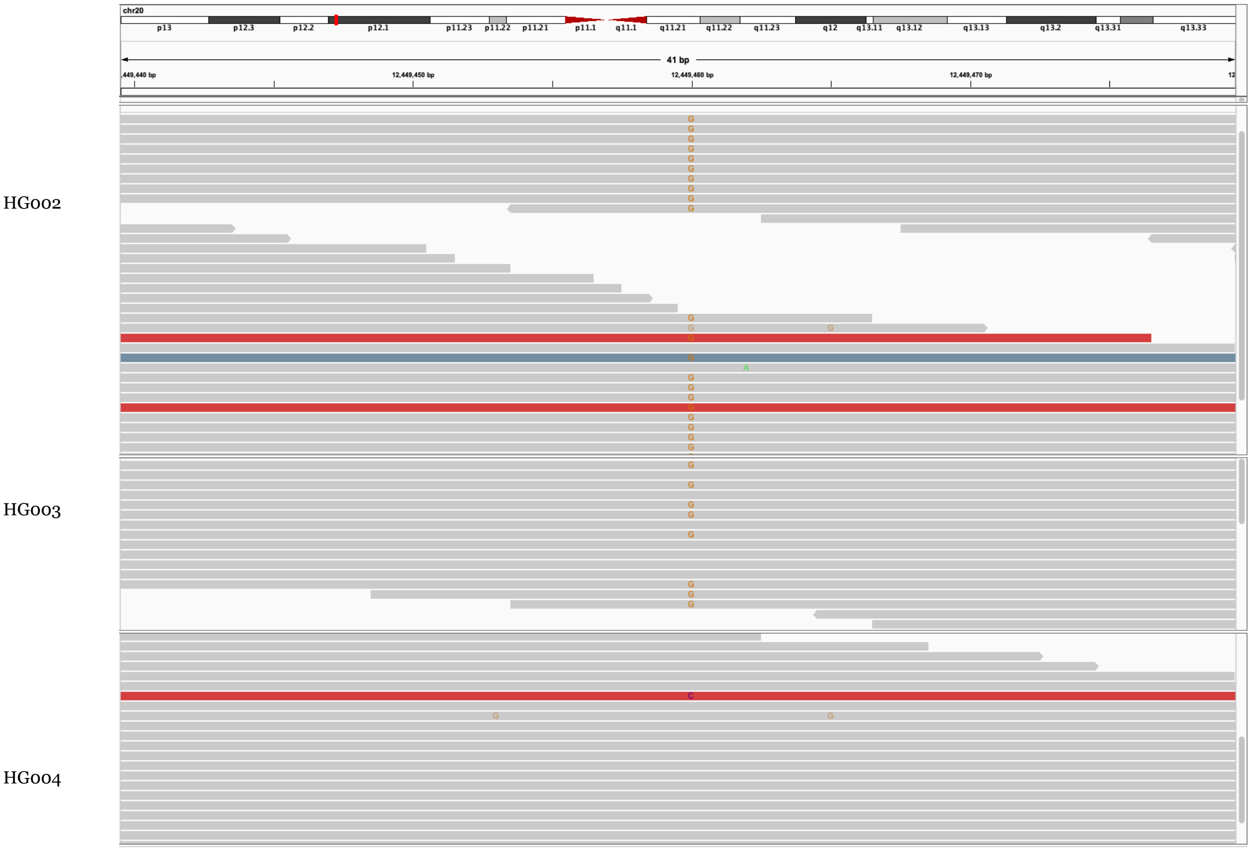
39. Chen, N.-C. *et al.* Improving variant calling using population data and deep learning. *Cold Spring Harbor Laboratory* 2021.01.06.425550 (2021) doi:10.1101/2021.01.06.425550.

40. Schneider, V. A. *et al.* Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Res.* **27**, 849–864 (2017).

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

42. Picard Tools - By Broad Institute. http://broadinstitute.github.io/picard/.

Supplementary Material



Supplementary Figure 1. Example of a variant called correctly by DeepTrio but not DeepVariant

IGV image of chr20:12449460 in 35x Illumina WGS PrecisionFDA v2 Truth Challenge samples for HG002-HG003-HG004. HG002 (child) is shown in the top row. This position is incorrectly genotyped as a homozygous variant in HG002 by DeepVariant and is correctly called a heterozygous variant in DeepTrio. This position is difficult to call because only a few reads are non-reference in HG002, and because some reads in this window are mapped discordantly.

Supplementary Table 1. SNP accuracy at 20x Coverage (all trio members)

Tool	Data Type	F1	Recall	Prec	FN	FP	FP. gt	TP de novo (all)	FP de novo
DeepTrio (trio)	Illumina	0.9970	0.9950	0.9991	359	66	25	27/34	13
DeepVariant	Illumina	0.9950	0.9930	0.9971	498	205	59	32/34	54
GATK4 (trio)	Illumina	0.9906	0.9923	0.9889	550	795	116	9/34	9
GATK4	Illumina	0.9902	0.9914	0.9890	612	790	132	34/34	202
dv-trio (trio)	Illumina	0.9946	0.9933	0.9960	481	285	76	0/34	0
Famseq (trio)	Illumina	0.9605	0.9807	0.9410	1374	4387	894	28/34	7
DeepTrio-PB	PacBio	0.9992	0.9989	0.9995	81	33	23	34	14
DeepVariant-PB	PacBio	0.9992	0.9986	0.9997	103	18	11	33/34	10

Supplementary Table 2. Indel accuracy at 20x Coverage (all trio members)

Tool	Data Type	F1	Recall	Prec	FN	FP	FP. gt	TP de novo (all)	FP de novo
DeepTrio (trio)	Illumina	0.9920	0.9890	0.9951	124	57	41	1/4	3
DeepVariant	Illumina	0.9854	0.9799	0.9910	226	104	74	1/4	15
GATK4 (trio)	Illumina	0.9824	0.9810	0.9836	213	191	99	1/4	2
GATK4	Illumina	0.9824	0.9794	0.9854	232	170	104	3/4	40
dv-trio (trio)	Illumina	0.9847	0.9788	0.9907	239	107	75	1/4	15
Famseq (trio)	Illumina	-	-	-	-	-	-	-	-
DeepTrio-PB	PacBio	0.9869	0.9881	0.9856	134	169	78	2/4	7
DeepVariant-PB	PacBio	0.9721	0.9702	0.9740	335	302	167	2/4	5

Commands Used

BWA

```
bwa mem -t 16 references/grch38_bwa_index/genome.fa
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.R1.fastq.gz
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.R2.fastq.gz -R
"@RG\\tID:${SAMPLE}\\tPL:ILLUMINA\\tPU:NONE\\tLB:${SAMPLE}\\tSM:${SAMPLE}" |
samtools sort -O BAM -o${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.grch38.bam
```

MarkDuplicates

```
java -jar gatk-package-4.1.2.0-local.jar MarkDuplicates -I
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.grch38.bam -O
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam -M
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.metrics
```

GATK HaplotypeCaller

```
java -jar gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar HaplotypeCaller -I
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam -R
references/GRCh38.no_alt_analysis_set.fa.gz -O
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.gatk.g.vcf -L chr1 -L
chr2 -L chr3 -L chr4 -L chr5 -L chr6 -L chr7 -L chr8 -L chr9 -L chr10 -L
chr11 -L chr12 -L chr13 -L chr14 -L chr15 -L chr16 -L chr17 -L chr18 -L chr19
-L chr20 -L chr21 -L chr22 -L chrX -L chrY --emit-ref-confidence GVCF
```

GATK CombineGVCF

```
java -jar gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar CombineGVCFs -V
HG002.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.gatk.g.vcf -V
HG003.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.gatk.g.vcf -V
HG004.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.gatk.g.vcf -R
references/GRCh38.no_alt_analysis_set.fa.gz -O
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.combine_gvcf.g.vcf.gz
```

GATK GenotypeGVCF

```
java -jar gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar GenotypeGVCFs -V
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.combine_gvcf.g.vcf.gz
 -R references/GRCh38.no_alt_analysis_set.fa.gz -O
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.genotype_gvcf.g.vcf.gz
```

GATK CalculateGenotypePosteriors

```
java -jar gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar
CalculateGenotypePosteriors -V
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.genotype_gvcf.g.vcf.gz
 -ped HG002-HG003-HG004.gatk.ped --skip-population-priors -O
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.calculate_genotype_posteriors
.g.vcf.gz
```

DeepVariant

```
sudo docker run \
     -v "${PWD}/input":"/input"   \
     -v "${PWD}/output":"/output"  \
     -v "${PWD}/reference":"/reference" \
     google/deepvariant:1.1.0 \
     /opt/deepvariant/bin/run_deepvariant \
     --model_type WGS \
     --call_variants_extra_args="use_openvino=true" \
     --ref /reference/GRCh38.no_alt_analysis_set.fa.gz \
     --reads /input/${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam
     --output_vcf
/output/${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deepvariant.vcf.
gz \
     --num_shards 16  \
     --intermediate_results_dir /output/intermediate_results_dir \
     --output_gvcf
/output/${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deepvariant.g.vc
f.gz
```

GLnexus

```
sudo docker run \
 -v "${PWD}/data":"/data" \
 quay.io/mlin/glnexus:v1.2.7 \
 /usr/local/bin/glnexus_cli \
 --config DeepVariantWGS \
 /data/HG002.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deepvariant.g.vcf.gz  \
 /data/HG003.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deepvariant.g.vcf.gz \
 /data/HG004.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deepvariant.g.vcf.gz \
 | bcftools view - | bgzip -c >
${PWD}/data/HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.grch38.deepvariant.co
hort.vcf.gz
```

DeepTrio

```
sudo docker run \
     -v "${PWD}/input":"/input"   \
     -v "${PWD}/output":"/output"  \
     -v "${PWD}/reference":"/reference" \
     gcr.io/deepvariant-docker/deeptrio:1.0.1rc \
     /opt/deepvariant/bin/deeptrio/run_deeptrio \
     --model_type WGS \
     --call_variants_extra_args="use_openvino=true" \
     --ref /reference/GRCh38.no_alt_analysis_set.fa.gz \
     --reads_child
/input/HG002.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam \
     --reads_parent1
/input/HG003.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam \
     --reads_parent2
/input/HG004.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.bam \
     --output_vcf_child
/output/HG002.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.vcf.gz \
     --output_vcf_parent1
/output/HG003.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.vcf.gz \
     --output_vcf_parent2
/output/HG004.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.vcf.gz \
     --sample_name_child 'HG002' \
     --sample_name_parent1 'HG003' \
     --sample_name_parent2 'HG004' \
     --num_shards $(nproc)  \
     --intermediate_results_dir /output/intermediate_results_dir \
     --output_gvcf_child
/output/HG002.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.g.vcf.gz \
     --output_gvcf_parent1
/output/HG003.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.g.vcf.gz \
       --output_gvcf_parent2
/output/HG004.novaseq.pcr-free.${COVERAGE}x.dedup.grch38.deeptrio.g.vcf.gz
```

dv-trio Famseq

```
./FamSeq vcf -vcfFile
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.grch38.deepvariant.cohort.vcf
-pedFile HG002-HG003-HG004.ped -output
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.grch38.dvtrio.vcf.gz
```

Octopus (non-trio)

```
./octopus/bin/octopus -R references/GRCh38.no_alt_analysis_set.fa -I
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.dedup.bam -o
```

```
${SAMPLE}.novaseq.pcr-free.${COVERAGE}x.vcf.gz --forest
octopus/resources/forests/germline.v0.7.2.forest --threads 16
```

Octopus (trio)

```
time ./octopus/bin/octopus -R references/GRCh38.no_alt_analysis_set.fa -I
HG002.novaseq.pcr-free.${COVERAGE}x.dedup.bam
HG003.novaseq.pcr-free.${COVERAGE}x.dedup.bam
HG004.novaseq.pcr-free.${COVERAGE}x.dedup.bam -o
HG002-HG003-HG004.novaseq.pcr-free.${COVERAGE}x.octopus_trio.vcf.gz -F HG003
-M HG004 --threads 16
```

Accuray Comparison with hap.py

```
sudo docker run -i \
  -v "${INPUT_DIR}":"/input" \
  -v "${OUTPUT_DIR}":"/output" \
  pkrusche/hap.py /opt/hap.py/bin/hap.py \
  /input/"${TRUTH_VCF}" \
  /output/output.vcf.gz \
  -f /input/"${TRUTH_BED}" \
  -r /input/"${REF}" \
  -o /output/happy.output \
  --engine=vcfeval
  -l "${EVAL_REGION}"
```

The truth VCF and BED files for Hap.py comparison are the v4.2.1 Truth Set from Genome in a Bottle:

ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio