nature biotechnology

A universal SNP and small-indel variant caller using deep neural networks

Ryan Poplin[1,2], Pi-Chuan Chang[2], David Alexander[2], Scott Schwartz[2], Thomas Colthurst[2], Alexander Ku[2], Dan Newburger[1], Jojo Dijamco[1], Nam Nguyen[1], Pegah T Afshar[1], Sam S Gross[1], Lizzie Dorfman[1,2], Cory Y McLean[1,2] & Mark A DePristo[1,2]

Despite rapid advances in sequencing technologies, accurately calling genetic variants present in an individual genome from billions of short, errorful sequence reads remains challenging. Here we show that a deep convolutional neural network can call genetic variation in aligned next-generation sequencing read data by learning statistical relationships between images of read pileups around putative variant and true genotype calls. The approach, called DeepVariant, outperforms existing state-of-the-art tools. The learned model generalizes across genome builds and mammalian species, allowing nonhuman sequencing projects to benefit from the wealth of human ground-truth data. We further show that DeepVariant can learn to call variants in a variety of sequencing technologies and experimental designs, including deep whole genomes from 10X Genomics and Ion Ampliseq exomes, highlighting the benefits of using more automated and generalizable techniques for variant calling.

Calling genetic variants from next-generation sequencing (NGS) data has proven challenging because NGS reads are not only errorful (with error rates from ~0.1–10%) but arise from a complex error process that depends on properties of the instrument, preceding data processing tools, and the genome sequence itself[1–5]. State-of-the-art variant callers use a variety of statistical techniques to model these error processes to accurately identify differences between the reads and the reference genome caused either by real genetic variants or by errors in the reads[3–6]. For example, the widely used GATK uses logistic regression to model base errors, hidden Markov models to compute read likelihoods, and naive Bayes classification to identify variants, which are then filtered to remove likely false positives using a Gaussian mixture model with hand-crafted features capturing common error modes[5]. These techniques allow the GATK to achieve high but still imperfect accuracy on the Illumina sequencing platform[3,4]. Generalizing these models to other sequencing technologies (for example, Ion Torrent[7,8]) has proven difficult due to the need to manually retune or extend these statistical models, which is problematic in an area with such rapid technological progress[1].

Here we describe a variant caller, called DeepVariant, that replaces the assortment of statistical modeling components with a single deep learning model. Deep learning is a machine learning technique applicable to a variety of domains, including image classification[9], translation[10], gaming[11,12] and the life sciences[13–16]. This toolchain (**Fig. 1**) begins by finding candidate single nucleotide polymorphisms (SNPs) and indels in reads aligned to the reference genome with high sensitivity but low specificity using standard, algorithmic preprocessing techniques. The deep learning model, using the Inception architecture[17], emits probabilities for each of the three diploid genotypes at a locus using a pileup image of the reference and read data around each candidate variant (**Fig. 1**). The model is trained using labeled true genotypes, after which it is frozen and can then be applied to novel sites or samples. In the following experiments, DeepVariant was trained on an independent set of samples or variants from those being evaluated.

The deep learning model was trained without specialized knowledge about genomics or next-generation sequencing, and yet it can learn to call genetic variants more accurately than state-of-the-art methods. When applied to the Platinum Genomes Project NA12878 data[18], DeepVariant produced a callset with better performance than the GATK when evaluated on the held-out chromosomes of the Genome in a Bottle ground-truth set (**Supplementary Figs. 1a** and **2**). For further validation, we sequenced 35 replicates of NA12878 using a standard whole-genome sequencing (WGS) protocol and called variants on 27 replicates using a GATK best-practices pipeline and DeepVariant using a model trained on the other eight replicates (Online Methods). DeepVariant produced more accurate results with greater consistency across a variety of quality metrics (**Supplementary Fig. 1b** and **Supplementary Notes 1**, **10** and **11**).

Like many variant calling algorithms, the GATK relies on a model that assumes read errors to be independent[5]. Though this has long been recognized as an invalid assumption[2], the true likelihood function that models multiple reads simultaneously is unknown[5,19,20]. Because DeepVariant presents an image of all of the reads relevant for a putative variant together, the convolutional neural network (CNN) is able to account for the complex dependence among the reads by virtue of being a universal approximator[21]. This manifests itself as a

[1]Verily Life Sciences, Mountain View, California, USA. [2]Google Inc., Mountain View, California, USA. Correspondence should be addressed to M.A.D. (mdepristo@google.com).

Received 15 December 2017; accepted 2 August 2018; published online 24 September 2018; doi:10.1038/nbt.4235

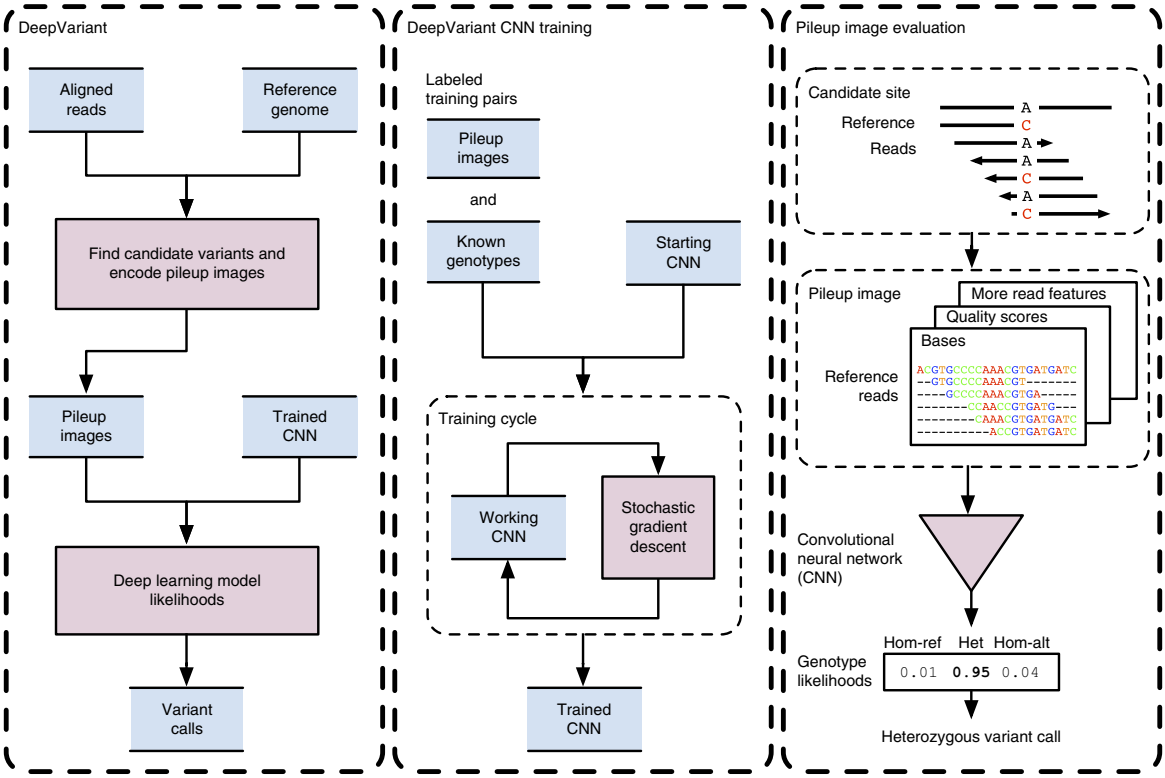

Figure 1 DeepVariant workflow overview. Before DeepVariant, NGS reads are first aligned to a reference genome and cleaned up with duplicate marking and, optionally, local assembly. Left box: first, the aligned reads are scanned for sites that may be different from the reference genome. The read and reference data are encoded as an image for each candidate variant site. A trained CNN calculates the genotype likelihoods for each site. A variant call is emitted if the most likely genotype is heterozygous or homozygous non-reference. Middle box: training the CNN reuses the DeepVariant machinery to generate pileup images for a sample with known genotypes. These labeled image + genotype pairs, along with an initial CNN, which can be a random model, a CNN trained for other image classification tests, or a prior DeepVariant model, are used to optimize the CNN parameters to maximize genotype prediction accuracy using a stochastic gradient descent algorithm. After a maximum number of cycles or time has elapsed or the model's performance has converged, the final trained model is frozen and can then be used for variant calling. Right box: the reference and read bases, quality scores, and other read features are encoded into a red–green–blue (RGB) pileup image at a candidate variant. This encoded image is provided to the CNN to calculate the genotype likelihoods for the three diploid genotype states of homozygous reference (hom-ref), heterozygous (het) or homozygous alternate (hom-alt). In this example a heterozygous variant call is emitted, as the most probable genotype here is "het". In all panels, blue boxes represent data and red boxes are processes. Details of all processes are given in the Online Methods.

tight concordance between the estimated probability of error from the likelihood function and the observed error rate (**Supplementary Fig. 1c**) where DeepVariant's CNN is well calibrated, more so than the GATK. That the CNN has approximated this true but unknown interdependent likelihood function is the essential technical advance enabling us to replace the hand-crafted statistical models used in other approaches with a single deep learning model, and still achieve such high performance in variant calling.

To further benchmark the performance of DeepVariant, we submitted variant calls for a blinded sample, NA24385, to the US Food and Drug Administration (FDA)-sponsored variant calling Truth Challenge in May 2016 and won the "highest performance" award for SNPs as assessed by an independent team using a different evaluation methodology. For this contest DeepVariant was trained only on data available from the CEPH (Centre d'Etude du Polymorphisme Humain) female sample NA12878 and was evaluated on the unseen Ashkenazi male sample NA24385. In achieving high accuracy as measured via F1, or the harmonic mean of sensitivity and positive predictive value (PPV), on this new sample (SNP F1 = 99.95%, indel F1 = 98.98%), we show that DeepVariant can generalize beyond its training data. We then applied the same dataset and evaluation methodology to a variety of both recent and commonly used bioinformatics methods, including the GATK, FreeBayes[22], SAMtools[23], 16GT[24] and Strelka[25] (**Table 1**). DeepVariant demonstrated more than 50% fewer errors per genome (4,652 errors) compared to the next-best algorithm (9,531 errors). We also evaluated the same set of methods using the synthetic diploid sample CHM1-CHM13[26] (**Table 2**). In our tests DeepVariant outperformed all other methods for calling both SNP and indel mutations, without needing to adjust filtering thresholds or other parameters.

We further explored how well DeepVariant's CNN generalizes beyond its training data. First, a model trained with read data aligned to human genome build GRCh37 and applied to reads aligned to GRCh38 had similar performance (overall F1 = 99.45%) to one trained on GRCh38 and then applied to GRCh38 (overall F1 = 99.53%), thereby demonstrating that a model learned from one version of the human genome reference can be applied to other versions with effectively no loss in accuracy (**Supplementary Table 1** and **Supplementary Note 2**). Second, models trained using human reads and ground-truth data achieved high accuracy when applied to a mouse dataset[27] (F1 = 98.29%), outperforming training on the mouse data itself (F1 = 97.84%; **Supplementary Table 2** and **Supplementary Note 3**).

Table 1 Evaluation of several bioinformatics methods on the high-coverage, whole-genome sample NA24385

Method	Type	F1	Recall	Precision	TP	FN	FP	FP.gt	FP.al	Version
DeepVariant (live GitHub)	Indel	0.99507	0.99347	0.99666	357,641	2350	1,198	217	840	Latest GitHub v0.4.1-b4e8d37d
GATK (raw)	Indel	0.99366	0.99219	0.99512	357,181	2810	1,752	377	995	3.8-0-ge9d806836
Strelka	Indel	0.99227	0.98829	0.99628	355,777	4214	1,329	221	855	2.8.4-3-gbe58942
DeepVariant (pFDA)	Indel	0.99112	0.98776	0.99450	355,586	4405	1,968	846	1,027	pFDA submission May 2016
GATK (VQSR)	Indel	0.99010	0.98454	0.99573	354,425	5566	1,522	343	909	3.8-0-ge9d806836
GATK (flt)	Indel	0.98229	0.96881	0.99615	348,764	11227	1,349	370	916	3.8-0-ge9d806836
FreeBayes	Indel	0.94091	0.91917	0.96372	330,891	29,100	12,569	9,149	3,347	v1.1.0-54-g49413aa
16GT	Indel	0.92732	0.91102	0.94422	327,960	32,031	19,364	10,700	7,745	v1.0-34e8f934
SAMtools	Indel	0.87951	0.83369	0.93066	300,120	59,871	22,682	2,302	20,282	1.6
DeepVariant (live GitHub)	SNP	0.99982	0.99975	0.99989	3,054,552	754	350	157	38	Latest GitHub v0.4.1-b4e8d37d
DeepVariant (pFDA)	SNP	0.99958	0.99944	0.99973	3,053,579	1,727	837	409	78	pFDA submission May 2016
Strelka	SNP	0.99935	0.99893	0.99976	3,052,050	3,256	732	87	136	2.8.4-3-gbe58942
GATK (raw)	SNP	0.99914	0.99973	0.99854	3,054,494	812	4,469	176	257	3.8-0-ge9d806836
16GT	SNP	0.99583	0.99850	0.99318	3,050,725	4,581	20,947	3,476	3,899	v1.0-34e8f934
GATK (VQSR)	SNP	0.99436	0.98940	0.99937	3,022,917	32,389	1,920	80	170	3.8-0-ge9d806836
FreeBayes	SNP	0.99124	0.98342	0.99919	3,004,641	50,665	2,434	351	1,232	v1.1.0-54-g49413aa
SAMtools	SNP	0.99021	0.98114	0.99945	2,997,677	57,629	1,651	1,040	200	1.6
GATK (flt)	SNP	0.98958	0.97953	0.99983	2,992,764	62,542	509	168	26	3.8-0-ge9d806836

The dataset used in this evaluation is the same as in the precisionFDA Truth Challenge (pFDA). Several methods are compared, including the DeepVariant callset as submitted to the contest and the most recent DeepVariant version from GitHub. Each method was run according to the individual authors' best-practice recommendations and represents a good-faith effort to achieve best results. Comparisons to the Genome in a Bottle truth set for this sample were performed using the hap.py software, available on GitHub at http://github.com/Illumina/hap.py, using the same version of the GIAB truth set (v3.2.2) used by pFDA. The overall accuracy (F1, sort order within each variant type), recall, precision, and numbers of true positives (TP), false negatives (FN) and false positives (FP) are shown over the whole genome. False positives are further divided by those caused by genotype mismatches (FP.gt) and those cause by allele mismatches (FP.al). Finally, the version of the software used for each method is provided. We present three GATK callsets: GATK (raw), the unfiltered calls emitted by the HaplotypeCaller; GATK (VQSR), the callset filtered with variant quality score recalibration (VQSR); and GATK (flt), the raw GATK callset filtered with run-flt in CHM-eval. See **Supplementary Note 7** for more details.

This last experiment is especially demanding as not only do the species differ but nearly all of the sequencing parameters do as well: 50× 2 × 148 bp from an Illumina TruSeq prep sequenced on a HiSeq 2500 for the human sample and 27× 2 × 100 bp reads from a custom sequencing preparation run on an Illumina Genome Analyzer II for mouse[27]. Thus, DeepVariant is robust to changes in sequencing depth, preparation protocol, instrument type, genome build and even mammalian species, thereby enabling resequencing projects in nonhuman species, which often have no ground-truth data to guide their efforts[27,28], to leverage the large and growing ground-truth data in humans.

To further assess its capabilities, we trained DeepVariant to call variants in eight datasets from Genome in a Bottle[29] that spanned a variety of sequencing instruments and protocols, including whole-genome and exome sequencing technologies, with read lengths from 50 to many thousands of base pairs (**Supplementary Tables 3** and **4** and **Supplementary Notes 4** and **5**). We used the already processed BAM files to introduce additional variability, as these BAMs differed in their alignment and cleaning steps. The results of this experiment all exhibit a characteristic pattern: the candidate variants have the highest sensitivity but a low PPV (mean of 57.6%), which varies substantially by dataset. After retraining, all of the callsets achieve high PPVs (mean of 99.3%) while largely preserving the candidate callset sensitivity (mean loss of 2.3%). The high PPVs and low loss of sensitivity indicate that DeepVariant can learn a model that captures the technology-specific error processes in sufficient detail to separate real variation from false positives with high fidelity for many different sequencing technologies.

Next we analyzed the behavior of DeepVariant on two non-Illumina WGS datasets, one from ThermoFisher (SOLiD) and one from Pacific Biosciences (PacBio), and on two exome datasets from Illumina (TruSeq) and Ion Torrent (Ion Ampliseq). The SOLiD and PacBio WGS datasets have high error rates in the candidate callsets. SOLiD (13.9% PPV for SNPs, 96.2% for indels and 14.3% overall) has many SNP artifacts from the mapping of short, color-space reads. The PacBio dataset is the opposite, with many false indels (79.8% PPV for SNPs, 1.4% for indels and 22.1% overall) owing to this technology's high indel error rate. Training DeepVariant to call variants in an exome is likely to be particularly challenging. Exomes have far fewer variants (~20k–30k)[30] than found in a whole genome (~4–5M)[31]. The non-uniform coverage and sequencing errors from the exome capture or amplification technology also introduce many false positive variants[32]. For example, at 8.1%, the PPV of our candidate variants for Ion Ampliseq is the lowest of all our datasets.

Despite the low initial PPVs, the retrained models in DeepVariant separated errors from real variants with high accuracy in the WGS datasets (PPVs of 99.0% and 97.3% for SOLiD and PacBio, respectively), though with a larger loss in sensitivity (candidates 82.5% and final 76.6% for SOLiD and 93.4% and 88.5%, respectively, for PacBio) than other technologies. Furthermore, despite the challenges of retraining deep learning models with limited data, the exome datasets also performed well, with a small reduction in sensitivity (from 91.9% to 89.3% and 94.0% to 92.6% for Ion Ampliseq and TruSeq candidates and final calls, respectively) for a substantial boost in PPV (from 8.1% to 99.7% and 65.3% to 99.3% for Ion and TruSeq, respectively). The performance of DeepVariant compares favorably to those of callsets submitted to the Genome in a Bottle project site using tools developed specifically for each NGS technology and to callsets produced by the GATK or SAMtools (**Supplementary Table 5**).

The accuracy numbers presented here should not be viewed as the maximum achievable by either the sequencing technology or DeepVariant. For consistency, we used the same model architecture, image representation, training parameters and candidate variant criteria for each technology. Because DeepVariant achieves high PPVs for all technologies, the overall accuracy is effectively driven by the sensitivity of the candidate callset. Improvements to the data processing steps before DeepVariant and the algorithm used to identify candidate variants is likely to translate into further improvements in overall accuracy, particularly for multi-allelic indels. Conversely, despite its

Table 2 Evaluation of several bioinformatics methods on the high-coverage, whole-genome synthetic diploid sample CHM1-CHM13

Method	Type	F1	Recall	Precision	TP	FN	FP	Version
DeepVariant	Indel	0.95806	0.92868	0.98936	529,137	40,634	5,690	v0.4.1-b4e8d37d
Strelka	Indel	0.95074	0.91623	0.98796	522,039	47,732	6,363	2.8.4-3-gbe58942
16GT	Indel	0.94010	0.90803	0.97452	517,369	52,402	13,527	v1.0-34e8f934
GATK (raw)	Indel	0.93268	0.89504	0.97363	509,969	59,802	13,811	3.8-0-ge9d806836
GATK (VQSR)	Indel	0.91212	0.84497	0.99087	481,441	88,330	4,437	3.8-0-ge9d806836
FreeBayes	Indel	0.90438	0.83025	0.99305	473,053	96,718	3,313	v1.1.0-54-g49413aa
SAMtools	Indel	0.86976	0.79089	0.96611	450,626	119,145	15,807	1.6
DeepVariant	SNP	0.99103	0.98888	0.99319	3,518,118	39,553	24,132	v0.4.1-b4e8d37d
Strelka	SNP	0.98865	0.98107	0.99636	3,490,314	67,357	12,749	2.8.4-3-gbe58942
16GT	SNP	0.97862	0.98966	0.96782	3,520,894	36,777	117,078	v1.0-34e8f934
FreeBayes	SNP	0.96910	0.94837	0.99075	3,373,984	183,687	31,492	v1.1.0-54-g49413aa
GATK (VQSR)	SNP	0.96895	0.94542	0.99368	3,363,476	194,195	21,379	3.8-0-ge9d806836
SAMtools	SNP	0.96818	0.94386	0.99378	3,357,947	199,724	21,012	1.6
GATK (raw)	SNP	0.96646	0.95685	0.97627	3,404,167	153,504	82,748	3.8-0-ge9d806836

Several methods are compared, including the most recent DeepVariant version from GitHub. Each method was run according to the individual authors' best-practice recommendations and represents a good faith effort to achieve best results. Comparisons to the CHM1-CHM13 truth set were performed using the CHM-eval.kit software, available on GitHub at https://github.com/lh3/CHM-eval, release version 0.5. The overall accuracy (F1, sort order within each variant type), recall, precision, and numbers of true positives (TP), false negatives (FN) and false positives (FP) are shown over the whole genome. Finally, the version of the software used for each method is provided. Note that we present two GATK callsets: GATK (raw), the unfiltered calls emitted by the HaplotypeCaller; and GATK (VQSR), the callset filtered with the VQSR. See **Supplementary Note 7** for more details.

effectiveness, representing variant calls as images and applying general image-classification models is certainly suboptimal, as we were unable to effectively encode all of the available information in the reads and reference into the three-channel image.

Taken together, our results demonstrate that the deep learning approach employed by DeepVariant can learn a statistical model describing the relationship between the experimentally observed NGS reads and genetic variants in that data for several sequencing technologies. Technologies like DeepVariant change the problem of calling variants from a process of expert-driven, technology-specific statistical modeling to a more automated process of optimizing a general model against data. With DeepVariant, creating an NGS caller for a new sequencing technology becomes a simpler matter of developing the appropriate preprocessing steps, training a deep learning model on sequencing data from samples with ground-truth data, and applying this model to new, even nonhuman, samples (see **Supplementary Note 6**).

At its core, DeepVariant generates candidate entities with high sensitivity but low specificity, represents the experimental data about each entity in a machine-learning-compatible format and then applies deep learning to assign meaningful biological labels to these entities. This general framework for inferring biological entities from raw, errorful, indirect experimental data is likely to be applicable to other high-throughput instruments.

The results presented in **Figure 1**, **Supplementary Figures 1** and **2**, and **Supplementary Tables 1–8** were generated with the original, internal version of DeepVariant. Since then we have rewritten DeepVariant to make it available as open source software. As a result, several improvements to the DeepVariant method have been made that are not captured in the analyses presented here, including switching to TensorFlow[33] to train the model, using the inception_v3 neural network architecture and using a multichannel tensor representation for the genomics data instead of an RGB image. The results in **Tables 1** and **2** used the open source version of DeepVariant; the evaluation scripts are available as **Supplementary Software**. The latest version of DeepVariant is available on GitHub (https://github.com/google/deepvariant/).

Also note that several other deep-learning-based variant callers have since been described[34,35].

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank J. Zook and his collaborators at NIST for their work developing the Genome in a Bottle resources, the Verily sequencing facility for running the NA12878 replicates, and our colleagues at Verily and Google for their feedback on this manuscript and the project in general. This work was supported by internal funding.

AUTHOR CONTRIBUTIONS

R.P. and M.A.D. designed the study, analyzed and interpreted results and wrote the paper. R.P., P.-C.C., D.A., S.S., T.C., A.K., D.N., J.D., N.N., P.T.A., S.S.G., L.D., C.Y.M. and M.A.D. performed experiments and contributed to the software.

COMPETING INTERESTS

D.N., J.D., N.N., P.T.A. and S.S.G. are employees of Verily Life Sciences. P.-C.C., D.A., S.S, T.C. and A.K. are employees of Google Inc. R.P., L.D., C.Y.M. and M.A.D. are employees of Verily Life Sciences and Google Inc. This work was internally funded by Verily Life Sciences and Google Inc.

1. Goodwin, S., McPherson, J.D. & McCombie, W.R. Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* **17**, 333–351 (2016).
2. Nielsen, R., Paul, J.S., Albrechtsen, A. & Song, Y.S. Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.* **12**, 443–451 (2011).
3. Li, H. Towards better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics* **30**, 2843–2851 (2014).
4. Goldfeder, R.L. *et al.* Medical implications of technical accuracy in genome sequencing. *Genome Med.* **8**, 24 (2016).
5. DePristo, M.A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491–498 (2011).
6. Ding, J. *et al.* Feature-based classifiers for somatic mutation detection in tumour-normal paired sequencing data. *Bioinformatics* **28**, 167–175 (2012).
7. Bragg, L.M., Stone, G., Butler, M.K., Hugenholtz, P. & Tyson, G.W. Shining a light on dark sequencing: characterising errors in Ion Torrent PGM data. *PLoS Comput. Biol.* **9**, e1003031 (2013).
8. Yeo, Z.X., Wong, J.C.L., Rozen, S.G. & Lee, A.S.G. Evaluation and optimisation of indel detection workflows for ion torrent sequencing of the BRCA1 and BRCA2 genes. *BMC Genomics* **15**, 516 (2014).
9. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process Syst.* **25**, 1097–1105 (2012).

10. Wu, Y. *et al.* Google's neural machine translation system: bridging the gap between human and machine translation. Preprint at https://arxiv.org/abs/1609.08144 (2016).
11. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
12. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
13. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.* **18**, 851–869 (2017).
14. Alipanahi, B., Delong, A., Weirauch, M.T. & Frey, B.J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
15. Zhou, J. & Troyanskaya, O.G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–934 (2015).
16. Xiong, H.Y. *et al.* The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347**, 1254806 (2015).
17. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. Preprint at https://arxiv.org/abs/1512.00567 (2015).
18. Eberle, M.A. *et al.* A reference dataset of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27**, 157–164 (2017).
19. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **18**, 1851–1858 (2008).
20. Li, R. *et al.* SNP detection for massively parallel whole-genome resequencing. *Genome Res.* **19**, 1124–1132 (2009).
21. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
22. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. Preprint at https://arxiv.org/abs/1207.3907 (2012).
23. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
24. 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).
25. Kim, S. *et al.* Strelka2: fast and accurate variant calling for clinical sequencing applications. Preprint at *bioRxiv* https://doi.org/10.1101/192872 (2017).
26. Li, H. *et al.* New synthetic-diploid benchmark for accurate variant calling evaluation. Preprint at *bioRxiv* https://doi.org/10.1101/223297 (2017).
27. Keane, T.M. *et al.* Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature* **477**, 289–294 (2011).
28. Van der Auwera, G. What are the standard resources for non-human genomes? http://gatkforums.broadinstitute.org/gatk/discussion/1243/what-are-the-standard-resources-for-non-human-genomes (2018).
29. Zook, J.M. *et al. Extensive Sequencing of Seven Human Genomes to Characterize Benchmark Reference Materials* (Cold Spring Harbor, 2015).
30. Lek, M. *et al.* Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536**, 285–291 (2016).
31. Auton, A. *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).
32. Robasky, K., Lewis, N.E. & Church, G.M. The role of replicates for error mitigation in next-generation sequencing. *Nat. Rev. Genet.* **15**, 56–62 (2014).
33. Abadi, M., Agarwal, A., Barham, P., Brevdo, E. & Chen, Z. TensorFlow: large-scale machine learning on heterogeneous systems, 2015. Preprint at https://arxiv.org/abs/1603.04467 (2015).
34. Luo, R., Sedlazeck, F.J., Lam, T.-W. & Schatz, M. Clairvoyante: a multi-task convolutional deep neural network for variant calling in single molecule sequencing. Preprint at *bioRxiv* https://doi.org/10.1101/310458 (2018).
35. Torracinta, R. & Campagne, F. Training genotype callers with neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/097469 (2016).

ONLINE METHODS

Haplotype-aware realignment of reads. Mapped reads are preprocessed using an error-tolerant, local De-Bruijn-graph-based read assembly procedure that realigns them according to their most likely derived haplotype. Candidate windows across the genome are selected for reassembly by looking for any evidence of possible genetic variation, such as mismatching or soft clipped bases. The selection criteria for a candidate window are very permissive so that true variation is unlikely to be missed. All candidate windows across the genome are considered independently. De Bruijn graphs are constructed using multiple fixed *k*-mer sizes (from 20 to 75, inclusive, with increments of 5) out of the reference genome bases for the candidate window, as well as all overlapping reads. Edges are given a weight determined by how many times they are observed in the reads. We trim any edges with weight less than three, except that edges found in the reference are never trimmed. Candidate haplotypes are generated by traversing the assembly graphs and the top two most likely haplotypes are selected that best explain the read evidence. The likelihood function used to score haplotypes is a traditional pair HMM with fixed parameters that do not depend on base quality scores. This likelihood function assumes that each read is independent. Finally, each read is then realigned to its most likely haplotype using a Smith–Waterman-like algorithm with an additional affine gap penalty score for homopolymer indels. This procedure updates both the position and the CIGAR string for each read.

Finding candidate variants. Candidate variants for evaluation with the deep learning model are identified with the following algorithm. We consider each position in the reference genome independently. For each site in the genome, we collect all the reads that overlap that site. The CIGAR string of each read is decoded and the corresponding allele aligned to that site is determined; these are classified into either a reference-matching base, a reference-mismatching base, an insertion with a specific sequence, or a deletion with a specific length. We count the number of occurrences of each distinct allele across all reads. See **Supplementary Note 8** and the current implementation at https://github.com/google/deepvariant/blob/r0.4/deepvariant/make_examples.py#L770.

If any candidates pass our calling thresholds at a site in the genome, we emit a VCF-like record with chromosome, start, reference bases and alternate bases, where reference bases and alternate bases are the VCF-compatible representation of all of the passing alleles.

We filter away any unusable reads (see is_usable_read() below) if a read is marked as a duplicate, if it is marked as failing vendor quality checks, if it is not aligned or is not the primary alignment, if its mapping quality is less than 10, or if it is paired and not marked as properly placed. We further only include read bases as potential alleles if all of the bases in the alleles have a base quality ≥10. We emit variant calls only at standard (ACGT) bases in the reference genome. It is possible to force candidate variants to be emitted (randomly with probability of *p*) at sites with no alternate alleles, which are used as homozygous reference training sites. There is no constraint on the size of indels emitted, so long as the exact position and bases are present in the CIGAR string and they are consistent across multiple reads.

Creating images around candidate variants. The second phase of DeepVariant encodes the reference and read support for each candidate variant into an RGB image. The pseudocode for this component is shown below; it contains all of the key operations to build the image, leaving out for clarity error handling, code to deal with edge cases such as those in which variants occur close to the start or end of the chromosome, and the implementation of nonessential and/or obvious functions. See **Supplementary Note 9** and the current implementation at https://github.com/google/deepvariant/blob/r0.4/deepvariant/pileup_image.py.

The actual implementation of this code uses a reservoir sampler to randomly remove reads at locations where there is excessive coverage. This downsampling occurs conceptually within the reads.get_overlapping() function but occurs in our implementation anywhere where there are more than 10,000 reads in a tiling of 300-bp intervals on the chromosome.

Deep learning. DistBelief[36] was used to represent models, train models on labeled images, export trained models, and evaluate trained models on unlabeled images. We adapted the Inception v2 architecture to our input images and our three-state (hom-ref, het, hom-alt) genotype classification problem. Specifically, we created an input image layer that rescales our input images to 299 × 299 pixels without shifting or scaling our pixel values. This input layer is attached to the ConvNetJuly2015v2[17] CNN with nine partitions and weight decay of 0.00004. The final output layer of the CNN is a three-class Softmax layer with fully connected inputs to the preceding layer initialized with Gaussian random weights and s.d. of 0.001 and a weight decay of 0.00004.

The CNN was trained using stochastic gradient descent in batches of 32 images with eight replicated models and RMS decay of 0.9. For the Platinum Genomes, precisionFDA, NA12878 replicates, mouse and genome build experiments, multiple models were trained (using the product of learning rates of [0.00095, 0.001, 0.0015] and momenta [0.8, 0.85, 0.9]) for 80 h or until training accuracy converged, and the model with the highest accuracy on the training set was selected as the final model. For the multiple sequencing technologies experiment, a single model was trained with learning rate 0.0015 and momentum 0.8 for 250,000 update steps. In all experiments unless otherwise noted, the CNN was initialized with weights from the ImageNet model ConvNetJuly2015v2[17].

DeepVariant inference client and allele merging. At inference time each biallelic candidate variant site represented as a pileup image is presented as input to the trained CNN. After a forward pass through the network, a three-state probability distribution is returned. These probabilities correspond to the biallelic genotype likelihood states of {*P*(homozygous reference), *P*(heterozygous), *P*(homozygous variant)} and are encoded directly in the output VCF record as the phred scaled GL field. Variant calls are emitted for all sites where the most likely genotype is either het or hom-alt with at least a Q4 genotype confidence. Finally, all biallelic records at the same starting position are merged into multiallelic records to facilitate comparisons with other datasets.

Genome in a Bottle human reference datasets. We used version 3.2.1 of the Genome in a Bottle reference data[37]. We downloaded calls in VCF format and confident called intervals in BED format from the following:

NA12878: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/NISTv3.2.1/

NA24385: https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/AshkenazimTrio/HG002_NA24385_son/NISTv3.2.1/

The VCF files were converted to Global Alliance for Global Health (GA4GH) protocol buffer format but otherwise were used without further modification.

Evaluating variant calls. An internal evaluation tool was used for some analyses. As of the availability of the open source version of DeepVariant, all analyses were completed using hap.py or CHM-eval (for example, see **Supplementary Tables 3** and **5**).

Truth variants and confident reference intervals were parsed from the Genome in a Bottle or other ground standard datasets from the VCF and BED files for their respective samples. Truth variants outside the confident intervals were removed. The evaluation variants were loaded and variants marked as filtered or assigned homozygous reference genotypes were removed. Metrics such as the number of SNPs, number of indels, insertion/deletion ratio, heterozygous/homozygous non-reference ratio and transition/transversion ratio (Ti/Tv) were calculated from all remaining evaluation variants.

Evaluation variants were matched to truth variants if they start at the same position on the same chromosome. To compute genotype concordance, we added to the list of matched pairs of evaluation–truth variants all of the unmatched evaluation variants that overlap the confidence intervals with a 'virtual' homozygous reference genotype sample. The number of matching genotypes is defined as the number of pairs in which the genotype alleles of the evaluation variant and truth variant are equal, independent of order. From this we compute the genotyping concordance as

$$\text{Genotype concordance} = \frac{\text{No. of matching genotypes}}{\text{No. of paired evaluation and truth variants}}$$

The number of matched pairs is counted as the number of true positives. Any truth variants without a matched evaluation variant are counted as false negatives. Any unmatched evaluation variants that occur within the confident intervals are

doi:10.1038/nbt.4235

counted as false positives. From the number of true positives (TP), false negatives (FN) and false positives (TP), we compute the sensitivity, PPV and F1 as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$\text{PPV} = \frac{\text{TP}}{\text{TP}+\text{FP}}$$

$$\text{F1} = \frac{2\text{TP}}{2\text{TP}+\text{FN}+\text{FP}}$$

Our evaluation metrics fall between the tolerant hapdip metric[3] and the strict vcfeval[38] metrics. In particular, our sensitivity and PPV metrics emphasize discriminating between variant and reference sites, allowing errors in the determination of the exact variant alleles and genotypes. These errors are tallied separately as an allelic error rate and a genotyping error rate. Although we believe this separation is informative and valuable for understanding the types of errors that occur in a variant callset, we appreciate the approaches pursued by other evaluation methods.

Life Sciences Reporting Summary. Further information about experimental design is available in the Nature Research Reporting Summary linked to this article.

Code availability. The latest version of DeepVariant is available at https://github.com/google/deepvariant. The key results and analyses presented here can be reproduced using the open-source version. Custom code was specific to our computing infrastructure and mainly used for simple data analysis tasks. The benchmarking script used to generate and evaluate the results in **Table 2** and **Supplementary Table 3** is available as **Supplementary Software**. An evaluation metrics file is available as **Supplementary Data**.

Data availability. All data used in this manuscript is publicly available from Genome in a Bottle or the Mouse Genome project, with the exception of 35 NA12878 WGS replicates from the Verily sequencing laboratory, which were licensed from Verily for the current study and are not publicly available. These data may be available from Verily upon reasonable request.

36. Dean, J. *et al.* Large scale distributed deep networks. *Adv. Neural Inf. Process. Syst.* **25**, 1223–1231 (2012).
37. 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).
38. Cleary, J.G. *et al.* Comparing variant call files for performance benchmarking of next-generation sequencing variant calling pipelines. Preprint at *bioRxiv* https://doi.org/10.1101/023754 (2015).

nature research

Corresponding author(s): Mark DePristo

Life Sciences Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form is intended for publication with all accepted life science papers and provides structure for consistency and transparency in reporting. Every life science submission will use this form; some list items might not apply to an individual manuscript, but all fields must be completed for clarity.

For further information on the points included in this form, see Reporting Life Sciences Research. For further information on Nature Research policies, including our data availability policy, see Authors & Referees and the Editorial Policy Checklist.

Please do not complete any field with "not applicable" or n/a. Refer to the help text for what text to use if an item is not relevant to your study.
For final submission: please carefully check your responses for accuracy; you will not be able to make changes later.

Experimental design

1. Sample size

   Describe how sample size was determined.

   No statistical claims based on sample sizes are made in the paper.

2. Data exclusions

   Describe any data exclusions.

   No data was excluded.

3. Replication

   Describe the measures taken to verify the reproducibility of the experimental findings.

   Multiple replicates of NA12878 were evaluated. Participated in a public blinded variant calling evaluation administered by a third-party (PrecisionFDA). Released the code to github, enabling additional third-party evaluations.

4. Randomization

   Describe how samples/organisms/participants were allocated into experimental groups.

   No randomization was needed.

5. Blinding

   Describe whether the investigators were blinded to group allocation during data collection and/or analysis.

   No blinding was needed.

Note: all in vivo studies must report how sample size was determined and whether blinding and randomization were used.

6. Statistical parameters

   For all figures and tables that use statistical methods, confirm that the following items are present in relevant figure legends (or in the Methods section if additional space is needed).

n/a	Confirmed	
☒	☐	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement (animals, litters, cultures, etc.)
☒	☐	A description of how samples were collected, noting whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	A statement indicating how many times each experiment was replicated
☒	☐	The statistical test(s) used and whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of any assumptions or corrections, such as an adjustment for multiple comparisons
☒	☐	Test values indicating whether an effect is present *Provide confidence intervals or give results of significance tests (e.g. P values) as exact values whenever appropriate and with effect sizes noted.*
☒	☐	A clear description of statistics including central tendency (e.g. median, mean) and variation (e.g. standard deviation, interquartile range)
☒	☐	Clearly defined error bars in all relevant figure captions (with explicit mention of central tendency and variation)

See the web collection on statistics for biologists for further resources and guidance.

Nature Biotechnology: doi:10.1038/nbt.4235

Software

Policy information about availability of computer code

7. Software

Describe the software used to analyze the data in this study.	A fully functional version of the software has been released to GitHub at https://github.com/google/deepvariant.

For manuscripts utilizing custom algorithms or software that are central to the paper but not yet described in the published literature, software must be made available to editors and reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). *Nature Methods* guidance for providing algorithms and software for publication provides further information on this topic.

Materials and reagents

Policy information about availability of materials

8. Materials availability

Indicate whether there are restrictions on availability of unique materials or if these materials are only available for distribution by a third party.	All samples analyzed here are available from Coriell as part of the Genome in a Bottle collections. Contact Verily re the availability of the 35 NA12878 replicates analyzed here.

9. Antibodies

Describe the antibodies used and how they were validated for use in the system under study (i.e. assay and species).	No antibodies were used.

10. Eukaryotic cell lines

a. State the source of each eukaryotic cell line used.	Coriell
b. Describe the method of cell line authentication used.	Genotyping concordance with public databases of genetic variants for these samples.
c. Report whether the cell lines were tested for mycoplasma contamination.	Unknown.
d. If any of the cell lines used are listed in the database of commonly misidentified cell lines maintained by ICLAC, provide a scientific rationale for their use.	None of the cell lines used are listed in the ICLAC database

Animals and human research participants

Policy information about studies involving animals; when reporting animal research, follow the ARRIVE guidelines

11. Description of research animals

Provide all relevant details on animals and/or animal-derived materials used in the study.	No animals were used in the study.

Policy information about studies involving human research participants

12. Description of human research participants

Describe the covariate-relevant population characteristics of the human research participants.	The study did not involve human research participants.

Nature Biotechnology: doi:10.1038/nbt.4235