nature genetics

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41588-018-0145-5

In the format provided by the authors and unedited.

Accurate genotyping across variant classes and lengths using variant graphs

Jonas Andreas Sibbesen[1,3], Lasse Maretty[1,3], The Danish Pan-Genome Consortium[2] and Anders Krogh[1]*

[1] The Bioinformatics Centre, Department of Biology, University of Copenhagen, Copenhagen, Denmark. [2] A complete list of consortium members is provided in the Supplementary Note. [3] These authors contributed equally: Jonas Andreas Sibbesen, Lasse Maretty. *e-mail: krogh@binf.ku.dk

Supplementary Information

Supplementary Note 2

The Danish Pan-Genome Consortium 2

Genotype inference 4

Supplementary Tables 7

Supplementary Table 1: Variation-prior sources 7

Supplementary Table 2: Computation time and memory usage for BayesTyper 7

Supplementary Table 3: Variant discovery workflow used on simulated and Platinum Genomes data 8

Supplementary Table 4: Re-genotyping workflow used on simulated, Platinum Genomes and Genome of the Netherlands data 8

Supplementary Note

The Danish Pan-Genome Consortium

Lasse Maretty[1], Jacob Malte Jensen[2,3], Bent Petersen[4], Jonas Andreas Sibbesen[1], Siyang Liu[1,5], Palle Villesen[2,3,6], Laurits Skov[2,3], Kirstine Belling[4], Christian Theil Have[7], Jose M. G. Izarzugaza[4], Marie Grosjean[4], Jette Bork-Jensen[7], Jakob Grove[3,8,9], Thomas D. Als[3,8,9], Shujia Huang[10,11], Yuqi Chang[10], Ruiqi Xu[5], Weijian Ye[5], Junhua Rao[5], Xiaosen Guo[10,12], Jihua Sun[5,7], Hongzhi Cao[10], Chen Ye[10], Johan v Beusekom[4], Thomas Espeseth[13,14], Esben Flindt[12], Rune M Friborg[2,3], Anders E. Halager[2,3], Stephanie Le Hellard[14,15], Christina M Hultman[16], Francesco Lescai[3,8,9], Shengting Li[3,8,9], Ole Lund[4], Peter Løngren[4], Thomas Mailund[2,3], Maria Luisa Matey-Hernandez[4], Ole Mors[3,6,9], Christian NS Pedersen[2,3], Thomas Sicheritz-Pontén[4], Patrick Sullivan[16,17], Ali Syed[4], David Westergaard[4], Rachita Yadav[4], Ning Li[5], Xun Xu[10], Torben Hansen[7], Anders Krogh[1], Lars Bolund[8,10], Thorkild IA Sørensen[7,18,19], Oluf Pedersen[7], Ramneek Gupta[4], Simon Rasmussen[4], Søren Besenbacher[2,6], Anders D. Børglum[3,8,9], Jun Wang[3,10,12], Hans Eiberg[20], Karsten Kristiansen[10,12], Søren Brunak[4,21], Mikkel Heide Schierup[2,3,22]

Affiliations

[1]Bioinformatics Centre, Department of Biology, University of Copenhagen, 2200 Copenhagen N, Denmark

[2]Bioinformatics Research Centre, Aarhus University, 8000 Aarhus C, Denmark

[3]iSEQ, Centre for Integrative Sequencing, Aarhus University, 8000 Aarhus C, Denmark

[4]DTU Bioinformatics, Department of Bio and Health Informatics, Technical University of Denmark, Kemitorvet, 2800 Kongens Lyngby, Denmark

[5]BGI-Europe, Ole Maaløes Vej 3, 2200 Copenhagen N, Denmark

[6]Department of Clinical Medicine, Aarhus University, 8000 Aarhus C, Denmark

[7]Novo Nordisk Foundation Center for Basic Metabolic Research, Section of Metabolic Genetics, University of Copenhagen, 2100 Copenhagen Ø, Denmark

[8]Department of Biomedicine, Aarhus University, 8000 Aarhus C, Denmark

[9]The Lundbeck Foundation Initiative for Integrative Psychiatric Research, iPSYCH, Denmark

[10]BGI-Shenzhen, Shenzhen 518083, China

[11]School of Bioscience and Biotechnology, South China University of Technology, Guangzhou 510006, China

[12]Laboratory of Genomics and Molecular Biomedicine, Department of Biology, University of Copenhagen, 2100 Copenhagen Ø, Denmark

[13]Department of Psychology, University of Oslo, Norway

[14]NORMENT, KG Jebsen Centre for Psychosis Research, Department of Clinical Science, University of Bergen, Bergen, 5021, Norway

[15]Dr E. Martens Research Group of Biological Psychiatry, Center for Medical Genetics and Molecular Medicine, Haukeland University Hospital, Bergen, 5021, Norway

[16]Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, 17177, Sweden

[17]Department of Genetics, University of North Carolina, Chapel Hill, NC 27599-7264, USA

[18]Department of Clinical Epidemiology (formerly Institute of Preventive Medicine), Bispebjerg and Frederiksberg Hospital, The Capital Region, Copenhagen, Denmark

[19]Department of Public Health, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark

[20]Department of Cellular and Molecular Medicine, University of Copenhagen, 2200 Copenhagen N, Denmark

[21]Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, 2200 Copenhagen N, Denmark

[22]Department of Bioscience, Aarhus University, 8000 Aarhus C, Denmark

Genotype inference

The inference objective is to estimate the posterior distribution over diplotypes, haplotype frequencies and count distribution parameters given a vector of k-mer counts for each individual in a population under the generative model described in the Methods. As k-mers not observed in any haplotype candidate carry abundant information about the genomic k-mer count distribution, we use counts from approximately 10 million of these k-mers to pre-estimate the negative binomial distribution parameters for each individual. The parameters are estimated using method-of-moments such that

$$p = \frac{\hat{\mu}}{\hat{\sigma}^2} \text{ and } r = \frac{\hat{\mu}^2}{\hat{\sigma}^2 - \hat{\mu}}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ denote the sample mean and variance, respectively[1]. We further pre-estimate the haplotype sparsity parameter π as the number of haplotypes required to explain all k-mers that are unique to the cluster divided by the total number of haplotypes as described previously by Maretty *et al.*[2]

The model assumes that k-mer counts within a cluster are independent, but they are in fact correlated due to sharing of reads and sequencing bias. Since the correlation decreases with the genomic distance between k-mers, we can reduce the impact of this correlation by randomly subsampling the k-mers used for inference. For each k-mer, we therefore sample whether it is to be used for inference from a Bernoulli distribution with a success probability of 0.1, resulting in a k-mer subset that is roughly 10 times smaller than the full set in the cluster. Furthermore, to reduce computation time in clusters containing large structural variation, the number of k-mers that overlaps an allele (reference and alternative) is limited to approximately 500 for each candidate haplotype after subsampling. Unique k-mers are prioritised over multi-cluster k-mers (observed in more than one cluster) in the latter step.

The posterior distribution over diplotypes, frequencies and noise parameters is then inferred using the following collapsed Gibbs sampling scheme. The sampler is initialised by drawing the Poisson parameter for each individual from the gamma prior with both the shape and scale parameters set to 1 and the haplotype frequencies for each cluster from the $|H|$-dimensional, symmetric Dirichlet distribution with a concentration parameter of 1. The algorithm proceeds as follows.

Step 1: Sampling of diplotypes

Independently for each individual, a diplotype *d* is drawn for each cluster conditioned on the individual's k-mer counts and count distribution parameters as well as the shared haplotype frequencies from the conditional posterior distribution given by

$$P(d|C,F,p,r,\lambda) = \frac{P(C|d,p,r,\lambda)P(d|F)}{\sum_{e\in D} P(C|e,p,r,\lambda)P(e|F)}$$

$P(d|F)$ denotes the multinomial distribution with probability parameters given by the frequencies *F* and the number of trials given by the ploidy, and

$$p(C|d,p,r,\lambda) = \prod_{i=1}^{n} P(C_i|m_i,p,r,\lambda)$$

where the product runs over all (subsampled) k-mers and m_i is the combined number of occurrences of the i'th k-mer in the diplotype d and the non-variable genomic regions.

Step 2: Sampling of frequencies

Independently for each cluster, a frequency vector *F* is drawn conditioned on the set of sampled individual diplotypes (step 1) and the sparsity parameter π using the sequential sampling scheme derived in Maretty *et al.*[2] In brief, the number of non-zero frequencies is first sampled conditioned on the sparsity parameter π and number of occurrences of each haplotype across the sampled diplotypes. Then, if more haplotypes than those that already had at least one diplotype occurrence are sampled to have positive frequency, these are selected randomly among the haplotypes that did not have a diplotype occurrence. Finally, the frequencies are drawn from the Dirichlet distribution conditioned on the sampled non-zero frequency indicators and the haplotype occurrences in the diplotypes across individuals.

Step 3: Sampling of noise parameters

Independently for each individual, the noise parameter λ is drawn conditioned on the noise k-mer counts c_n and gamma prior from the gamma posterior distribution:

$$P(\lambda|c_n,k,\theta) = Gamma\left(k + \sum c_n, \frac{\theta}{n_n\theta + 1}\right)$$

where n_n is the total number of noise k-mers ($m = 0$) across all clusters and $\sum c_n$ is the sum of the counts of these n_n k-mers. The shape *k* and scale θ parameters of the gamma prior were set to one.

Collapsed Gibbs sampling scheme

To reduce computation time, we first estimate the noise model parameters by running 200 iterations of the Gibbs sampler on 1,000,000 randomly chosen autosomal single nucleotide polymorphisms containing no ambiguous bases, no excluded k-mers and no k-mers occurring in non-variable regions or other clusters. We then fix the noise parameters to the mean of the last 100 iterations for all clusters and estimate their posterior distribution over genotypes by iteratively sampling individual diplotypes (step 1) and haplotype frequencies (step 2) (i.e. without sampling the noise parameters). The clusters in a group are sampled sequentially by traversing the group cluster tree (Supplementary Figure 14) using a depth-first search in each iteration, each time estimating the diplotype of the cluster dependent on the diplotype estimated in the outer cluster (i.e. at the parent node in the tree). To handle local minima arising from unfavourable subsampling of k-mers, we run 20 independent Gibbs sampling chains and in each of them we sample a new subset of k-mers. The sampling order of clusters that share the same parent node is also randomised in each Gibbs sampling chain. For each chain, we then run 100 burn-in iterations, followed by the collection of 250 samples yielding a total of 5,000 Gibbs samples used for estimation of the posterior distribution over diplotypes. Finally, the genotype posterior estimates are obtained by summing the posterior probabilities of diplotypes that specify the same genotype across all Gibbs sampling chains.

Multi-cluster k-mers

k-mers that are observed in more than one variant cluster within the same group are dependent (k-mers observed in different groups are excluded). To handle these, rather than using the k-mer multiplicity m defined above, we further include the multiplicity contributed by the current diplotype state of the other clusters in the group such that:

$$m = m_d + m_{vg} + m_g$$

where m_{vg} specifies the combined number of occurrences of the k-mer in the last sampled diplotypes of the other clusters in the group (i.e. $m_d + m_{vg}$ is equal to the total number of occurrences within the group). Multi-cluster k-mers are not used for inference in the first iteration since no sampled diplotypes are available for the other clusters at that time.

References

1. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biol.* **11,** R106 (2010).
2. Maretty, L., Sibbesen, J. A. & Krogh, A. Bayesian transcriptome assembly. *Genome Biol.* **15,** 501 (2014).

Supplementary Tables

Source	Version	GRCh37	GRCh38	Filter†
dbSNP	150	Yes	Yes	No rare SNVs
1000 Genomes project (1000G)	Phase 3	Yes	Yes	No SNVs
Genome of the Netherlands project (GoNL)	Release 6	Yes	Lifted*	No SNVs
Genotype-Tissue Expression project (GTEx)	Analysis V6	Yes	Lifted*	No SNVs
GenomeDenmark project (GDK)	v1.0	Lifted*	Yes	No SNVs

Supplementary Table 1: Variation-prior sources. *The coordinates of variants only called on one reference genome build were lifted over using *Crossmap*. †Reference and alternative alleles containing ambiguous nucleotides were removed from all sources.

Dataset	Number of samples	Max allele length (nts)	Number of variant alleles	Wall time* (hours)	Max memory (GB)
Simulation, 10x	10	500,000	14.6M	17	152
Simulation, 30x	10	500,000	13.4M	19	148
Platinum Genomes (PG)	13	500,000	11.7M	91	169
Platinum Genomes (PG)	13	10,000	11.7M	42	129
Platinum Genomes (PG)	13	500,000	61.1M	125	375
Genome of the Netherlands project (GoNL)	10 (children)	500,000	21.4M	16	159
Genome of the Netherlands project (GoNL)	10 (children)	500,000	64.4M	61	291

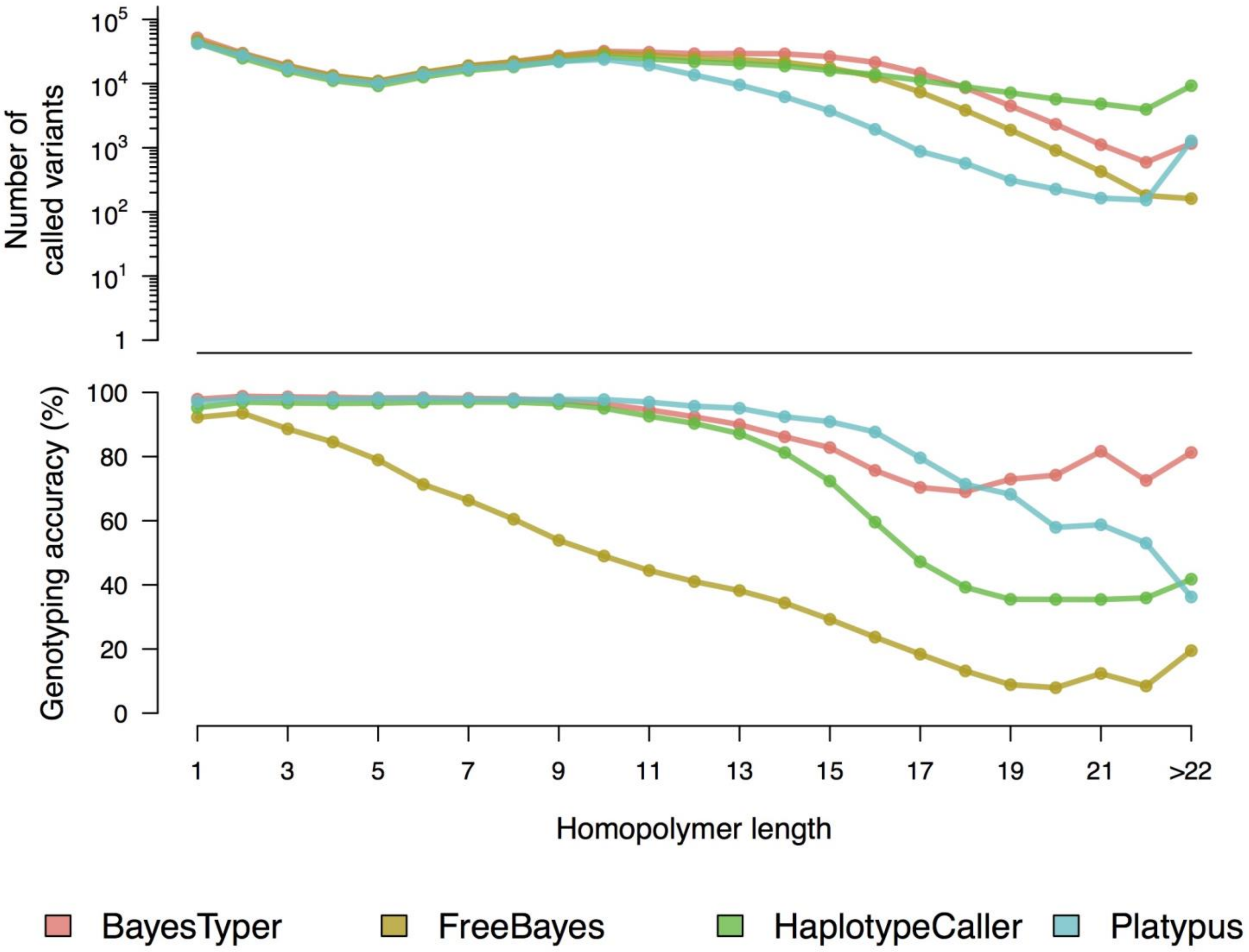
Supplementary Table 2: Computation time and memory usage for BayesTyper. *All runs were done on a 64-bit Intel Xeon 2.30 GHz machine with 1TB of memory using 32 threads.

Method	Options	Filters
FreeBayes (v1.1.0)	Default	None
HaplotypeCaller (v3.6 & v3.7)*	(-stand_emit_conf 10)†	None
Platypus (v0.8.1)	--assemble=1 --assembleBrokenPairs=1	None
Manta (v1.0.3)	Default	IMPRECISE, INS & TRA

Supplementary Table 3: Variant discovery workflow used on simulated and Platinum Genomes data. IMPRECISE: Variants without breakpoint resolution, INS: Insertions without sequence content, TRA: Translocation. *Version 3.6 was used on simulated data and version 3.7 on real data. †The *-stand_emit_conf* option was only used when running on simulated data (deprecated in *v3.7)*.

Method	Options	Filters
BayesTyper (v1.2)	Default	QUAL < 20, Het / NAK / FAK‡, GPP < 0.99
FreeBayes (v1.1.0)	--only-use-input-alleles --genotype-qualities	QUAL < 20, QUAL / AO <= 10, SAF == 0, SAR == 0, RPR <= 1, RPL <= 1, GQ < 20
HaplotypeCaller (v3.6 & v3.7)*	--genotyping_mode GENOTYPE_GIVEN_ALLELES --output_mode EMIT_ALL_SITES -stand_call_conf 0 (-stand_emit_conf 0)†	QUAL < 20, VQSRTranche >= 0.99, GQ < 20
Platypus (v0.8.1)	--minPosterior=0 --getVariantsFromBAMs=0	QUAL < 20, FILTER = PASS, GQ < 20
SVTyper (v0.1.4)	Default	QUAL < 20, GQ < 20

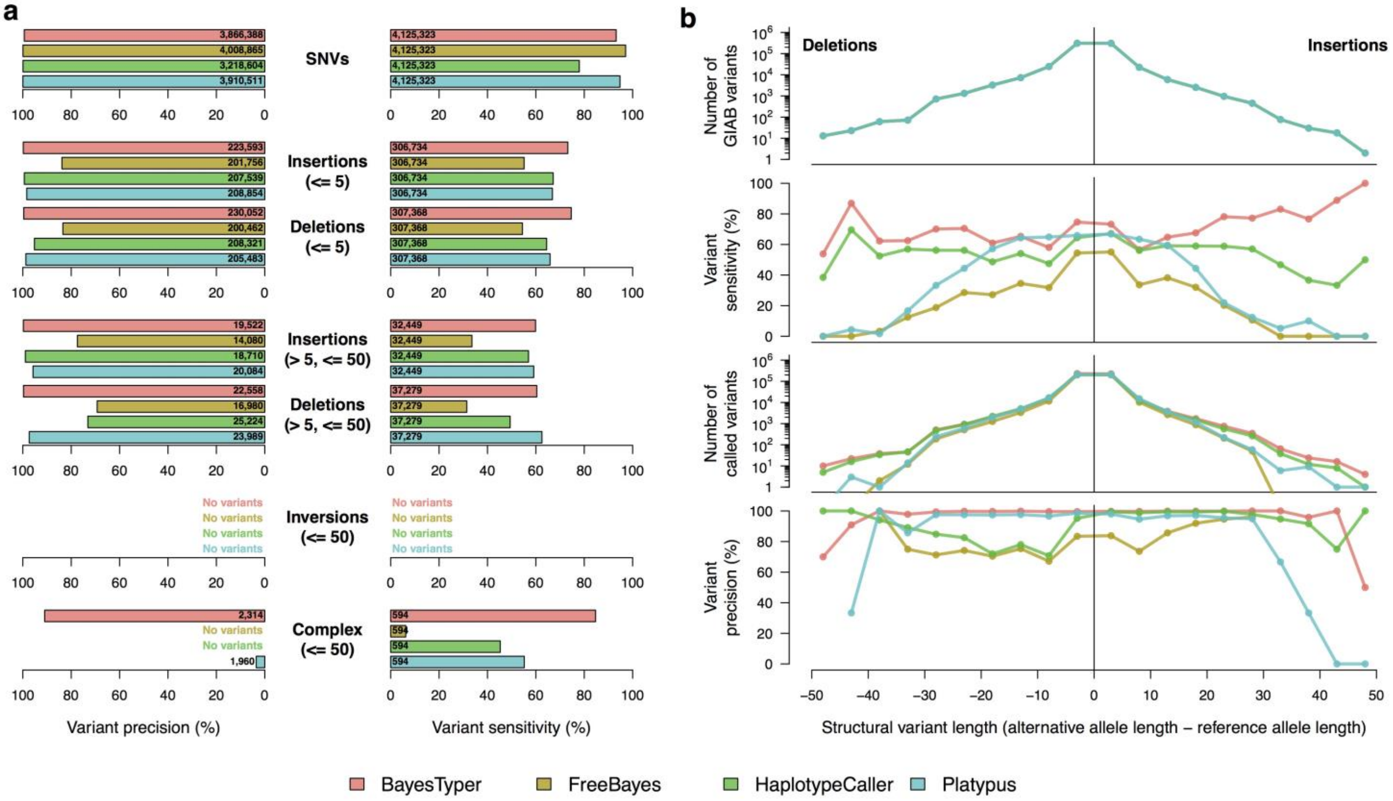
Supplementary Table 4: Re-genotyping workflow used on simulated, Platinum Genomes and Genome of the Netherlands data. *Version 3.6 was used on simulated data and version 3.7 on real data. †The *-stand_emit_conf* option was only used when running on simulated data (deprecated in *v3.7)*. ‡See Filtering section in the Methods for details on the BayesTyper filters.



Supplementary Figure 1

Comparison of homopolymer genotyping performance across methods on Platinum Genomes data (50×).

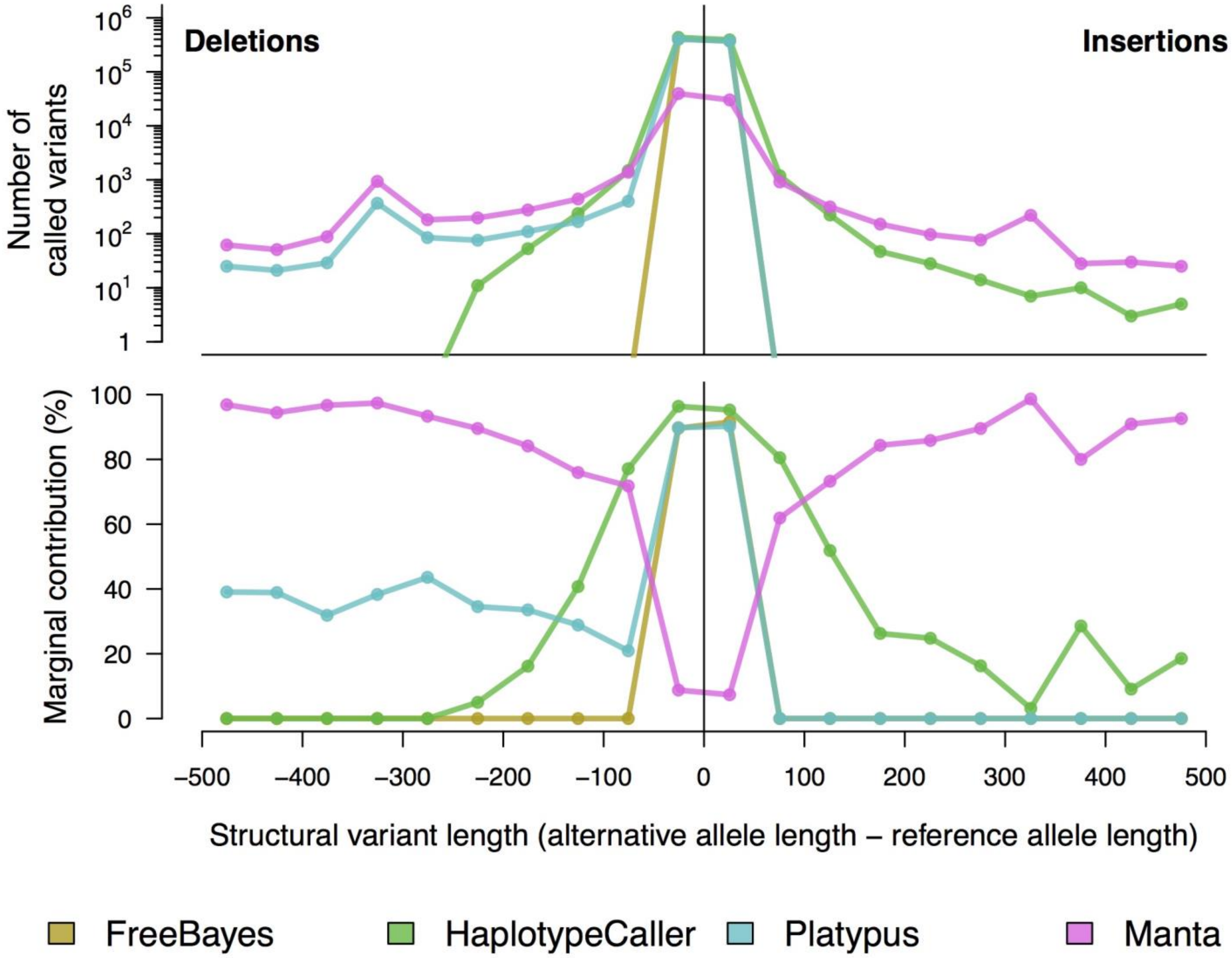
BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from 13 individuals in the Platinum Genomes pedigree using variants discovered by merging calls from four different methods as variant candidate input (Table 1). **a**,**b**, The number of called variant alleles (i.e., variant sensitivity) (**a**) and genotyping accuracy (**b**) estimated by validating genotypes using pedigree inheritance information shown as a function of the reference homopolymer length.



Supplementary Figure 2

Comparison of genotyping methods on Platinum Genomes data (50×) against the Genome in a Bottle 'ground-truth' set for NA12878.

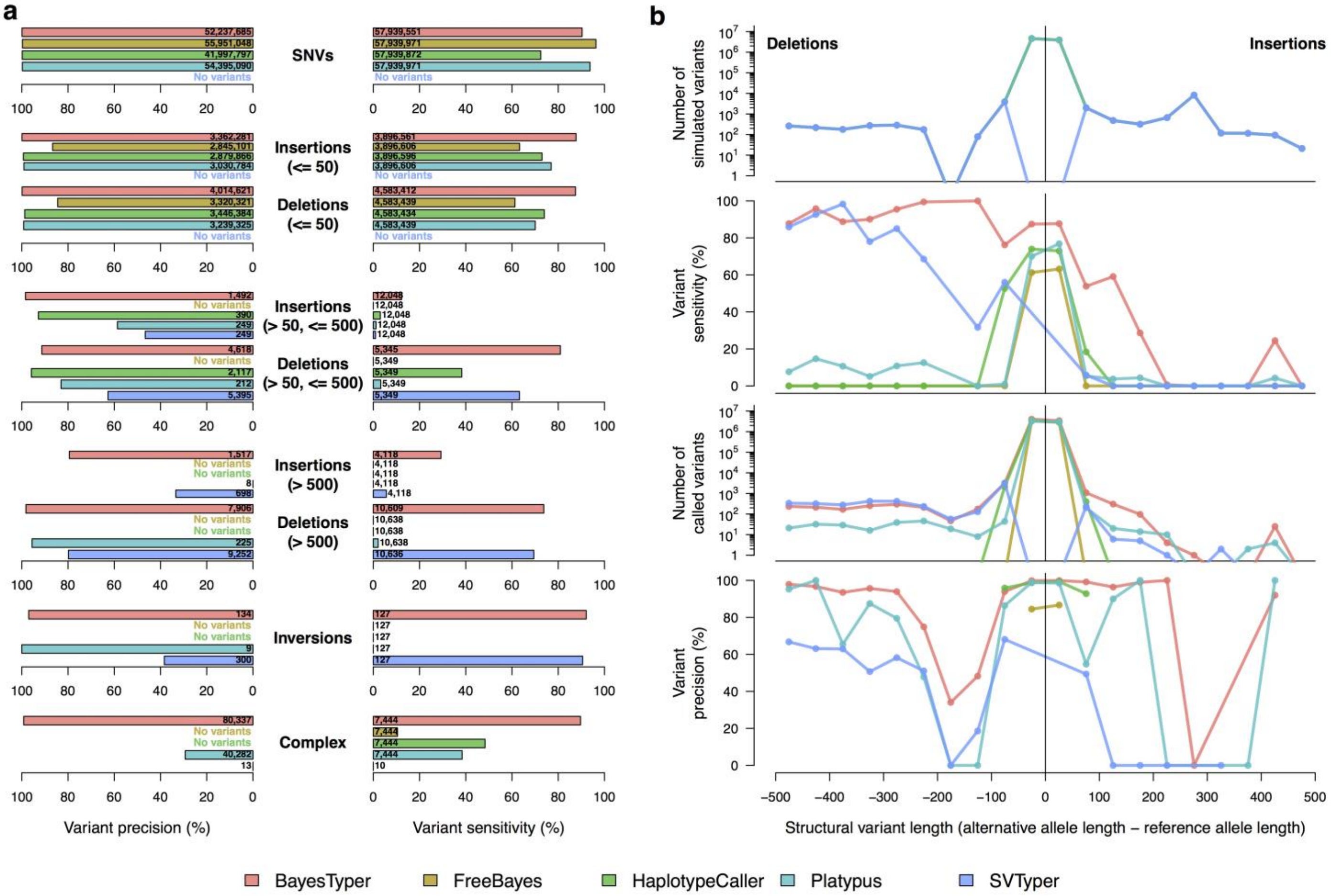
BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from 13 individuals in the Platinum Genomes pedigree using variants discovered by merging calls from four different methods as variant candidate input (Table 1). Variants longer than 50 nt were excluded from the analyses. **a**, Variant allele sensitivity (right) and precision (left) across variant classes; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. **b**, Sensitivity (two upper panels) and precision (two lower panels) for structural variants as a function of the net change in sequence length relative to the reference (5-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 3

Marginal contribution of discovery methods to the Platinum Genomes (50×) genotypes.

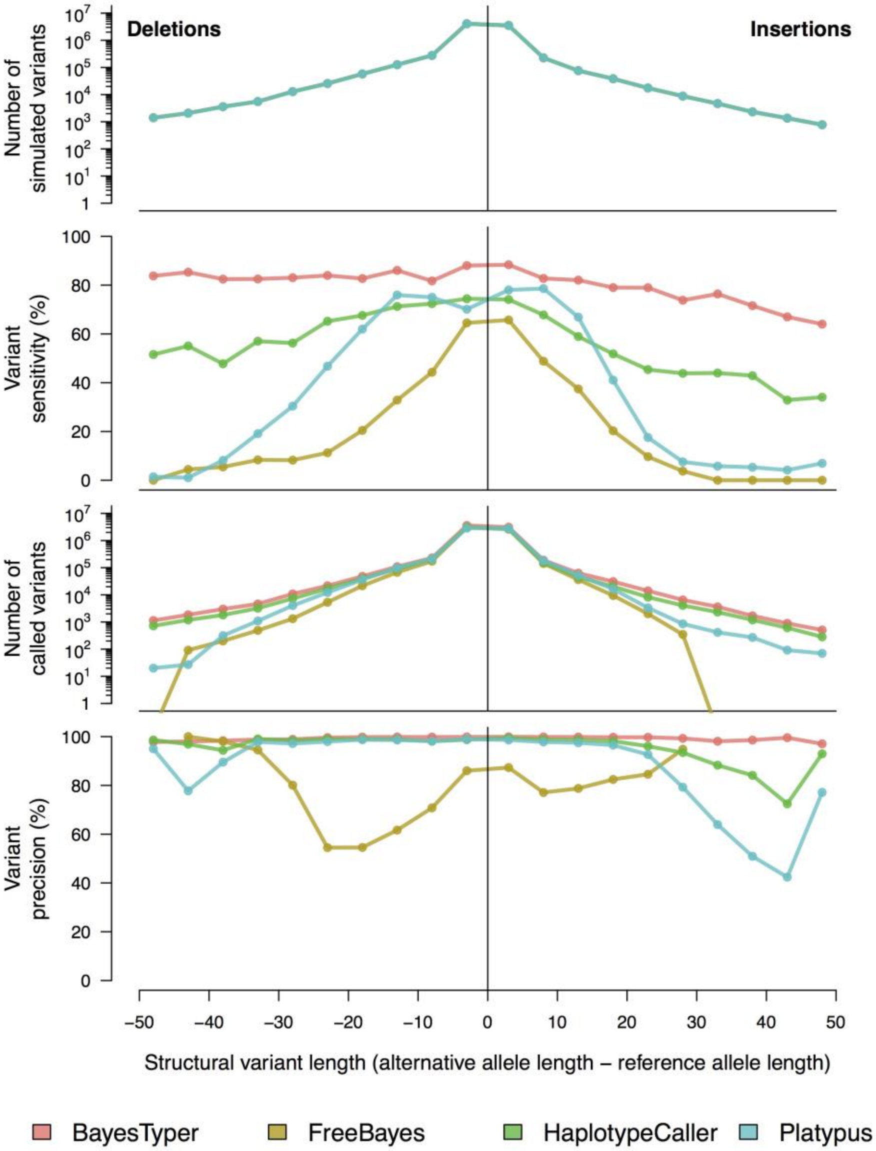
Variant discovery was conducted by merging calls from HaplotypeCaller, Platypus, FreeBayes and Manta across 13 individuals in the Platinum Genomes pedigree (Table 1). Plots show the absolute number of variants called by BayesTyper for each discovery method (top) and the marginal fraction of all called variants identified by each discovery method (bottom) for structural variants as a function of the net change in sequence length relative to the reference (50-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 4

Comparison of genotyping methods on simulated data (30×).

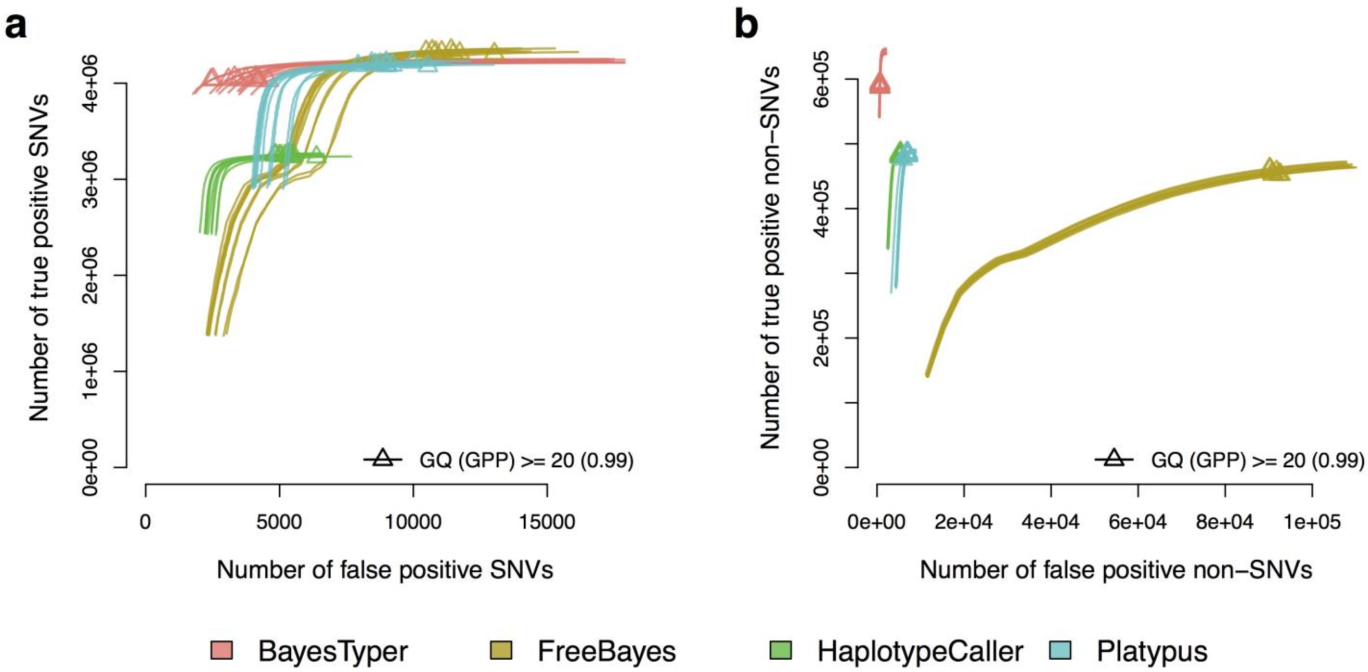
We simulated 30× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, SVTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as variant candidate input (Table 1). Values were aggregated across all ten individuals to provide a single estimate. **a**, Variant allele sensitivity (right) and precision (left) across variant classes; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. **b**, Sensitivity (two upper panels) and precision (two lower panels) for structural variants as a function of the net change in sequence length relative to the reference (50-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 5

Comparison of indel genotyping performance across methods on simulated data (30×).

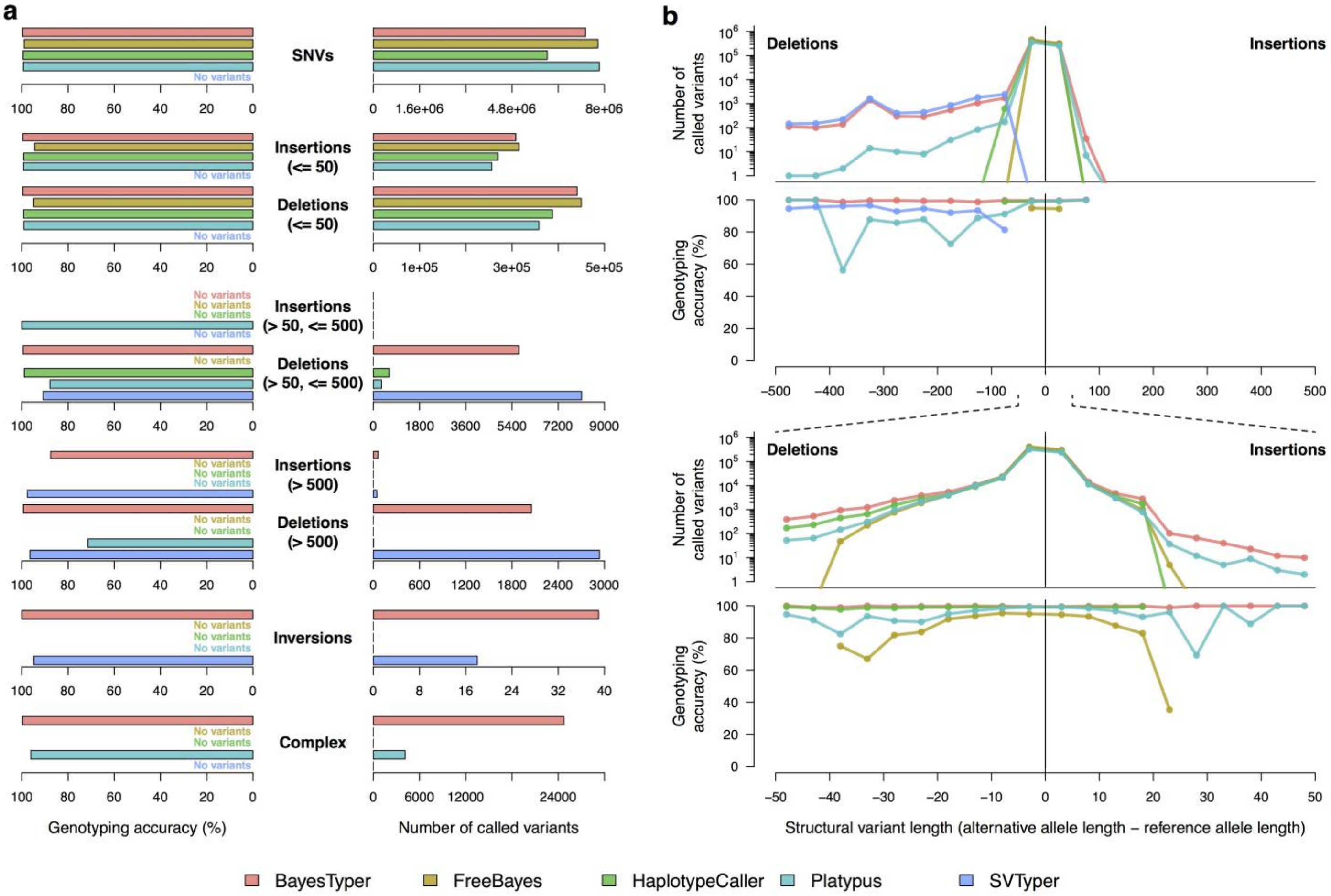
We simulated 30× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as variant candidate input (Table 1). Values were aggregated across all ten individuals to provide a single estimate. Plots show sensitivity (upper two panels) and precision (lower two panels) for indels as a function of the net change in sequence length relative to the reference (5-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 6

Receiver-operator curves on simulated data (30×).

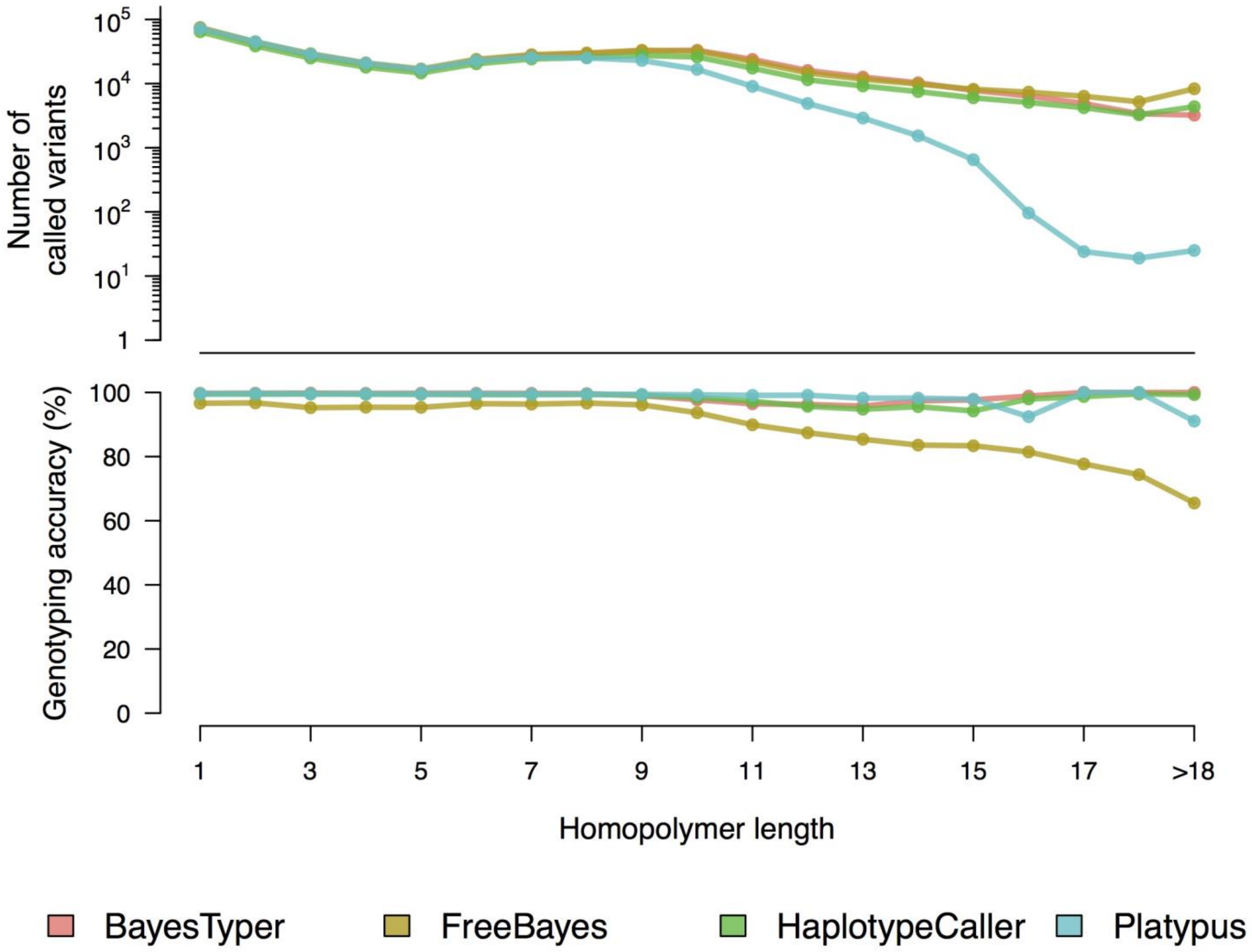
We simulated 30× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as variant candidate input (Table 1). **a**,**b**, Receiver-operator curves were computed for genotype quality (posterior probability for BayesTyper) across all methods for each of the ten simulated individuals for SNVs (**a**) and non-SNVs (**b**). Triangles indicate the genotype quality threshold used in the benchmark.



Supplementary Figure 7

Comparison of genotyping methods on Genome of the Netherlands data (13×).

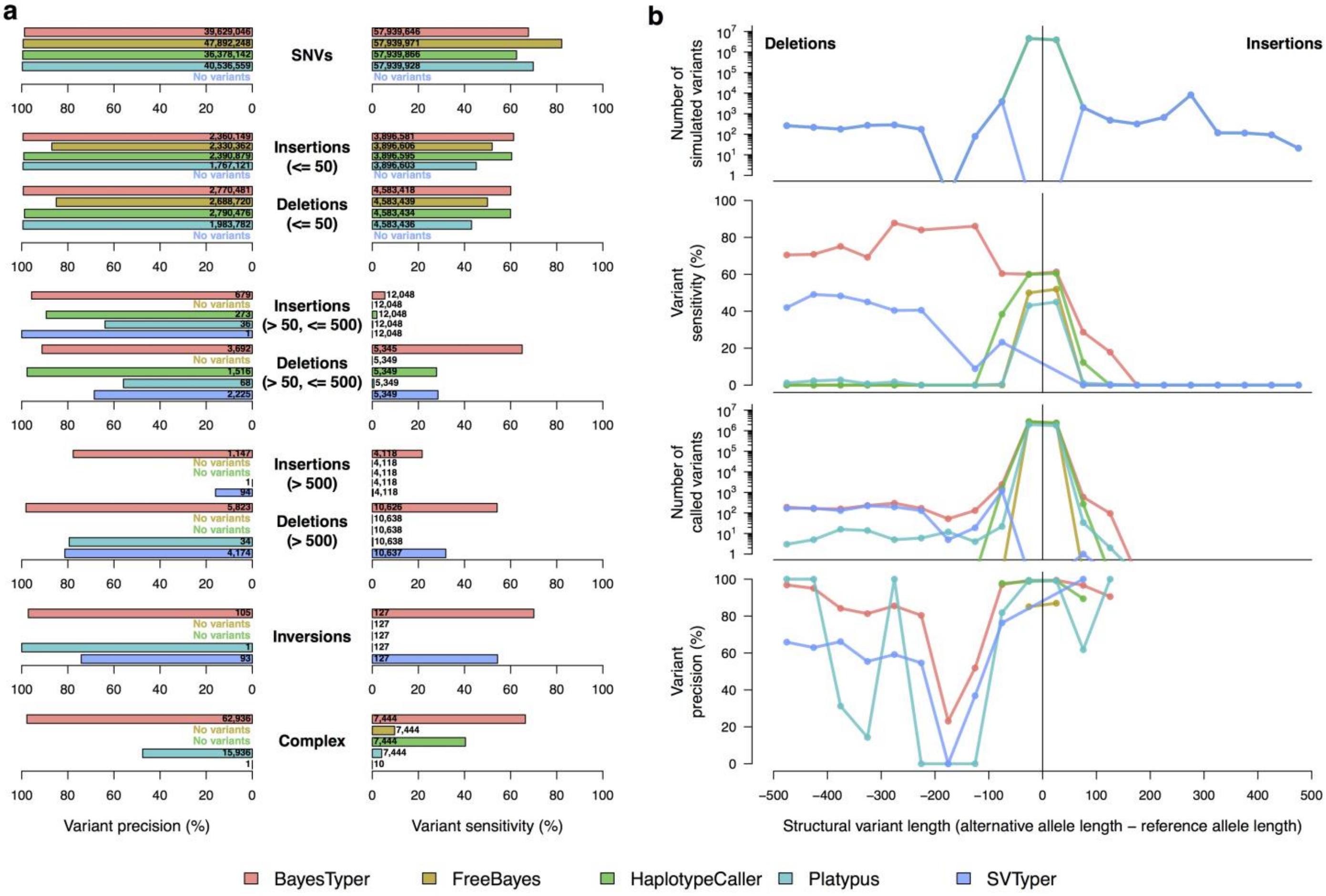
BayesTyper, SVTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from 30 individuals (ten parent–offspring trios) from the Genome of the Netherlands (GoNL) project using variant calls for the entire GoNL cohort (*n* = 769) obtained from the GoNL project as variant candidate input (Table 1). The number of called variant alleles was used as a measure of sensitivity, whereas genotyping accuracy was estimated as the fraction of variants with no Mendelian errors across the ten trios. **a**, Sensitivity (right) and accuracy (left) across variant classes; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. **b**, Sensitivity (top, log scale) and accuracy (bottom) for structural variants as a function of the net change in sequence length relative to the reference (50-nt and 5-nt bins for the ±500-nt and ±50-nt scales, respectively). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 8

Comparison of homopolymer genotyping performance across methods on Genome of the Netherlands data (13×).

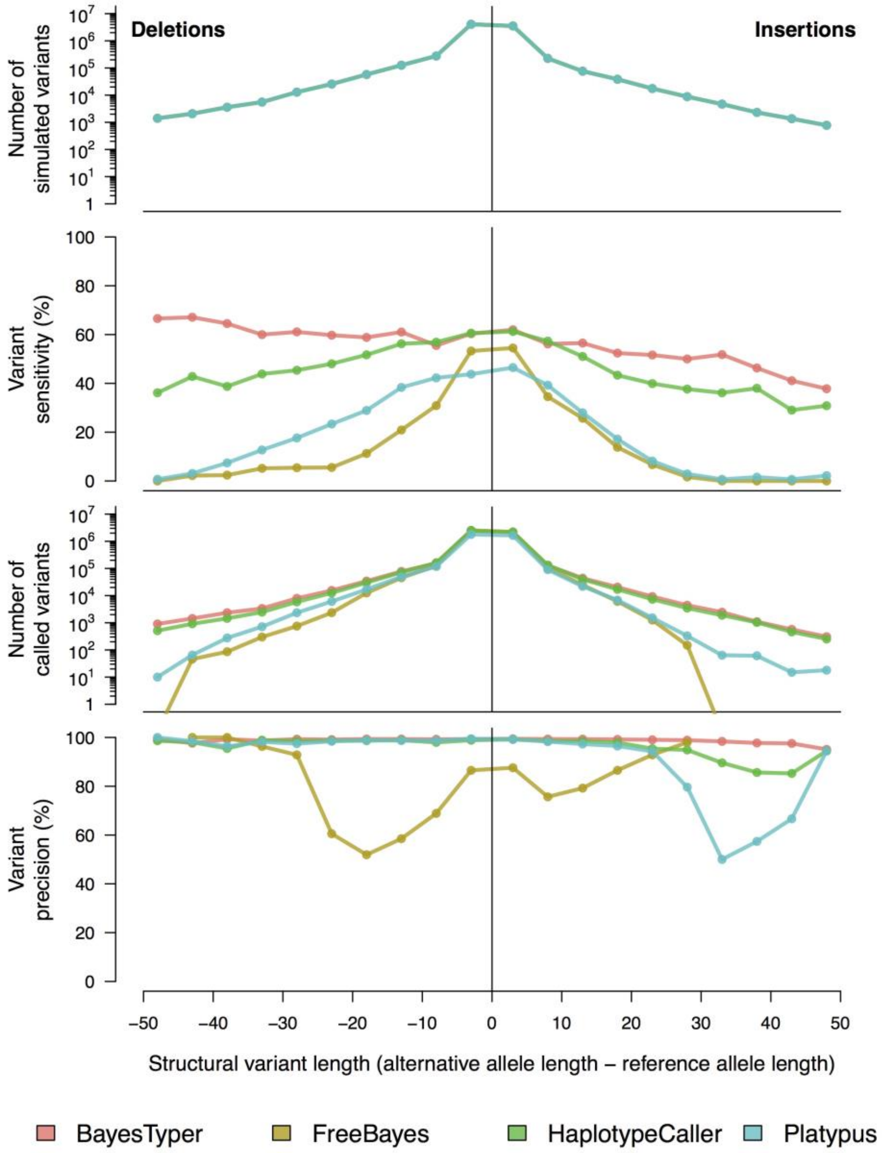
BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from 30 individuals (ten parent–offspring trios) from the Genome of the Netherlands (GoNL) project using variant calls for the entire GoNL cohort (*n* = 769) obtained from the GoNL project as variant candidate input (Table 1). **a,b**, The number of called variant alleles (i.e., variant sensitivity) (**a**) and genotyping accuracy (**b**) assessed by the fraction of variants with no Mendelian errors across the ten trios shown as a function of the reference homopolymer length.



Supplementary Figure 9

Comparison of genotyping methods on simulated data (10×).

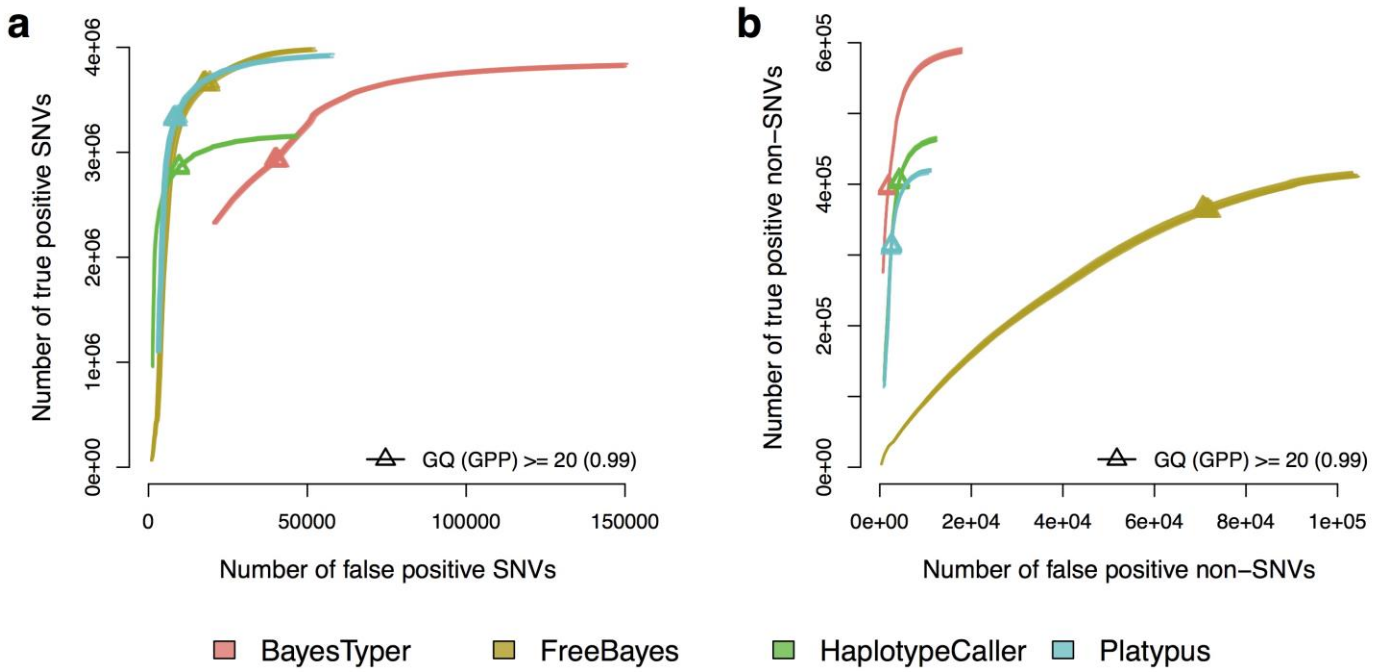
We simulated 10× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, SVTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as variant candidate input (Table 1). Values were aggregated across all ten individuals to provide a single estimate. **a**, Variant allele sensitivity (right) and genotyping accuracy (left) across variant classes; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. **b**, Sensitivity (upper two panels) and precision (lower two panels) for structural variants as a function of the net change in sequence length relative to the reference (50-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 10

Comparison of indel genotyping performance across methods on simulated data (10×).

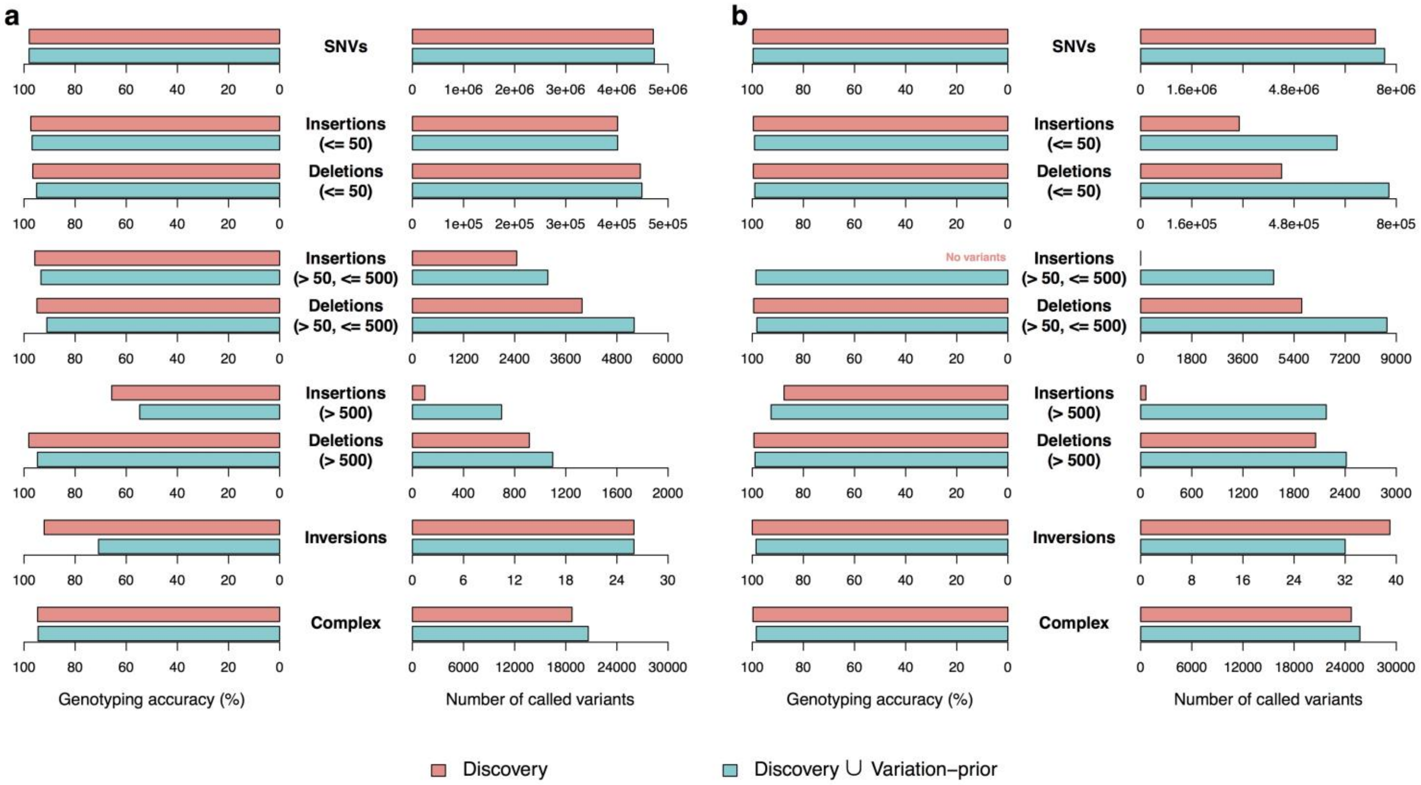
We simulated 10× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as variant candidate input (Table 1). Values were aggregated across all ten individuals to provide a single estimate. Plots show sensitivity (upper two panels) and precision (lower two panels) for indel variants as a function of the net change in sequence length relative to the reference (5-nt bins). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 11

Receiver-operator curves on simulated data (10×).

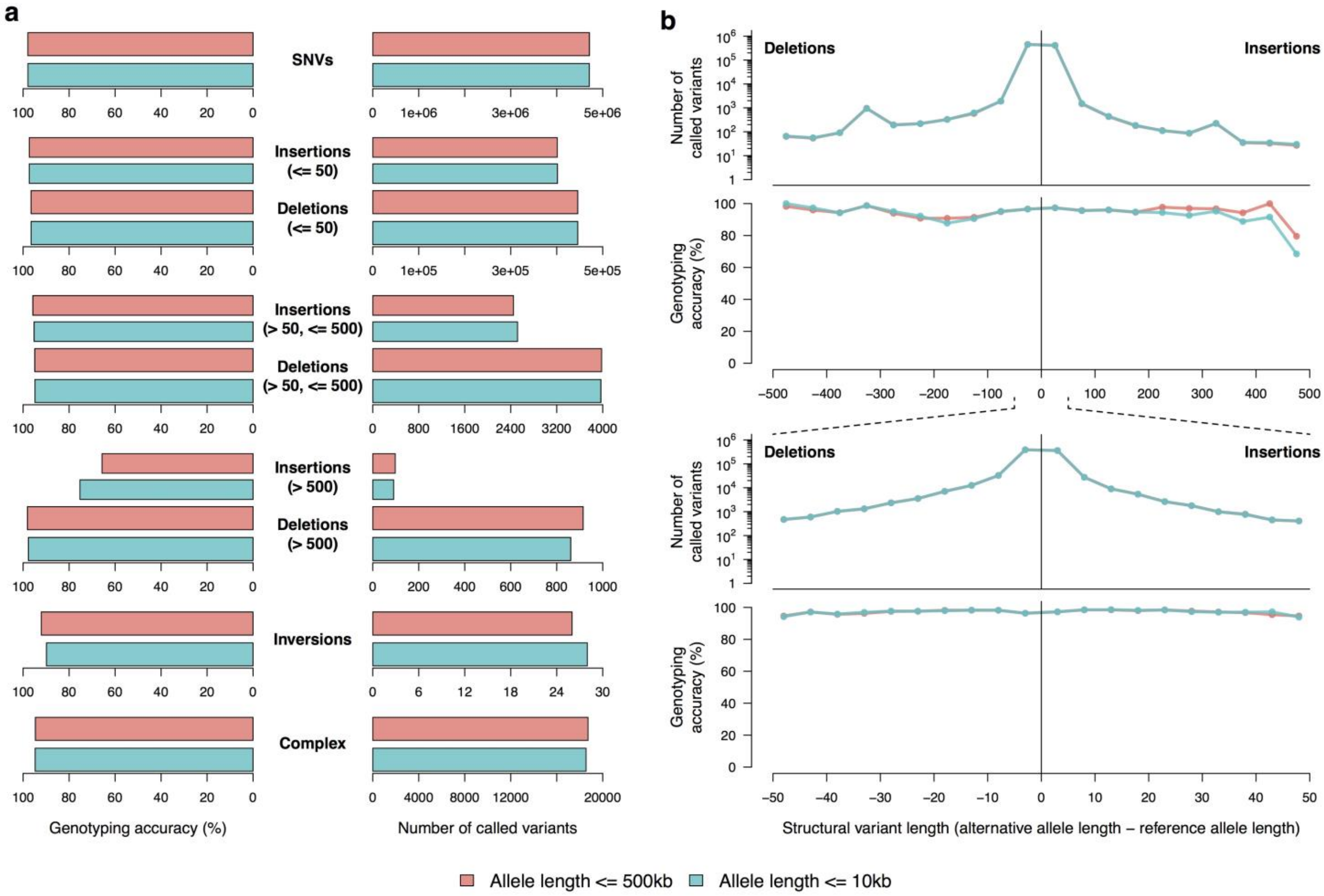
We simulated 10× paired-end sequencing data for ten Yoruba individuals from Ibadan, Nigeria (YRI) based on their 1000 Genomes genotype estimates. BayesTyper, HaplotypeCaller, Platypus and FreeBayes were run on data from the ten simulated individuals using variants discovered by merging calls from four different methods as input (Table 1). **a**,**b**, Receiver-operator curves were computed for genotype quality (posterior probability for BayesTyper) across all methods for each of the ten simulated individuals for SNVs (**a**) and non-SNVs (**b**). Triangles indicate the genotype quality threshold used in the benchmark.



Supplementary Figure 12

Effect of using BayesTyper with a variation prior on genotyping performance across variant classes on the Platinum Genomes (50×) and Genome of the Netherlands (13×) datasets.

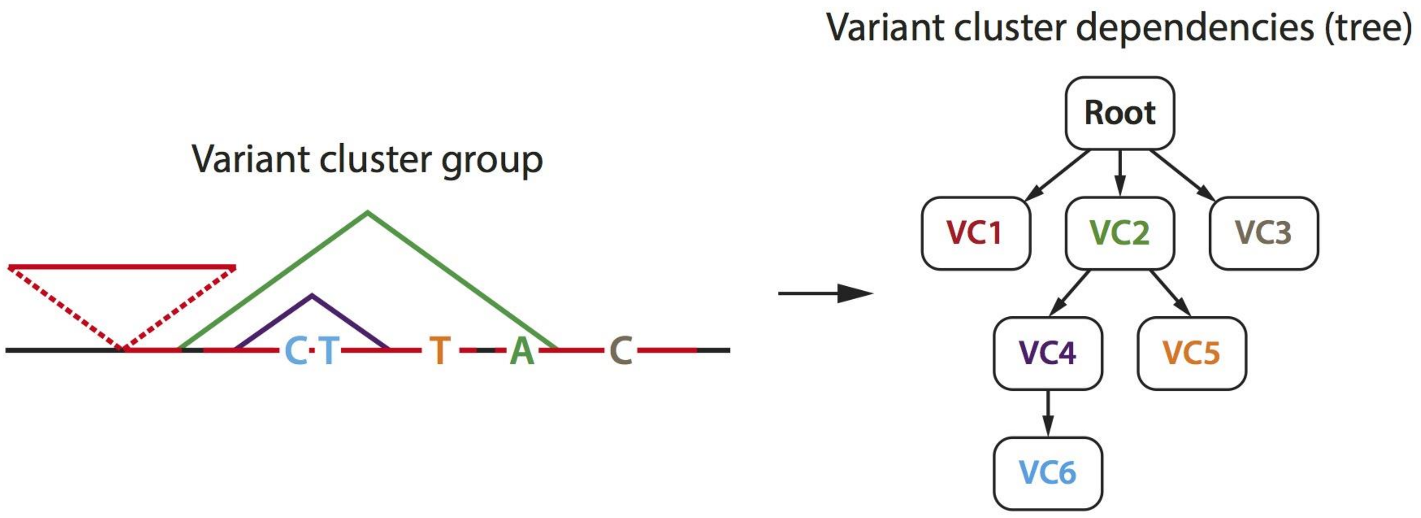
A ‘variation prior’ database was constructed by combining SNVs and structural variants from different databases and studies (Supplementary Table 1). BayesTyper was then run on variant candidates obtained by merging the variation prior with variants discovered using four different methods (Table 1). **a**, Sensitivity (right) and accuracy (left) of BayesTyper across variant classes on the Platinum Genomes (50×) datasets with and without the variation prior; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. Genotyping accuracy was estimated by validating genotypes using pedigree inheritance information. **b**, Same analyses as in **a** when running BayesTyper on the Genome of the Netherlands data (13×), where genotyping accuracy was estimated as the fraction of variants with no Mendelian errors across the ten trios.



Supplementary Figure 13

Effect of changing the ‘maximum allele length’ threshold in BayesTyper on genotyping performance across variant classes on the Platinum Genomes data (50×).

BayesTyper was run on data from 13 individuals in the Platinum Genomes pedigree using variants discovered by merging calls from four different methods as variant candidate input (Table 1) and using two different thresholds for the maximum allele length (10,000 and 500,000 nt). The number of called variant alleles was used as a measure of sensitivity, whereas the genotyping accuracy was estimated by validating genotypes using pedigree inheritance information. **a**, Sensitivity (right) and accuracy (left) across variant classes; variants not classified as SNVs, insertions, deletions or inversions were labeled as complex. **b**, Sensitivity (top, log scale) and accuracy (bottom) for structural variants as a function of the net change in sequence length relative to the reference (50-nt and 5-nt bins for the ±500-nt and ±50-nt scales, respectively). Variant alleles that do not entail a net change in sequence length (e.g., SNVs) were omitted.



Supplementary Figure 14

Variant cluster group definition.

Variant clusters within the copied or deleted sequence of an upstream structural variant are dependent, as they will share *k*-mers. These clusters are therefore defined to belong to the same inference group. Their dependency structure is represented as a tree, where the colors correspond to different variant clusters (VC); green and purple triangles are deletions, and the red triangle is a copy number insertion.