nature biotechnology

ARTICLES

https://doi.org/10.1038/s41587-019-0201-4

Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype

Daehwan Kim[1*], Joseph M. Paggi[2], Chanhee Park[1], Christopher Bennett[1] and Steven L. Salzberg[3,4]

The human reference genome represents only a small number of individuals, which limits its usefulness for genotyping. We present a method named HISAT2 (hierarchical indexing for spliced alignment of transcripts 2) that can align both DNA and RNA sequences using a graph Ferragina Manzini index. We use HISAT2 to represent and search an expanded model of the human reference genome in which over 14.5 million genomic variants in combination with haplotypes are incorporated into the data structure used for searching and alignment. We benchmark HISAT2 using simulated and real datasets to demonstrate that our strategy of representing a population of genomes, together with a fast, memory-efficient search algorithm, provides more detailed and accurate variant analyses than other methods. We apply HISAT2 for HLA typing and DNA fingerprinting; both applications form part of the HISAT-genotype software that enables analysis of haplotype-resolved genes or genomic regions. HISAT-genotype outperforms other computational methods and matches or exceeds the performance of laboratory-based assays.

Advances in sequencing technologies and computational methods have enabled rapid and accurate identification of genetic variants in the human population. Detailed individual genomic data, together with clinical and environmental information, promise to help improve predictions of cancer risk, inform lifestyle choices, generate more accurate clinical diagnoses, reduce adverse drug reactions (and other side effects of treatments) and improve patient outcomes through better-targeted therapies. Although massive sequencing projects over the past decade such as the 1000 Genomes Project[1,2], GTEx[3], GEUVADIS[4,5] and the Simons Simplex Collection (SSC)[6,7] have generated trillions of reads available from public archives[8], the ability to make use of these enormous datasets is still limited.

One important limitation is that most analyses rely on the alignment of sequencing reads to the human reference genome[9], which does not reflect the genetic diversity of individuals or populations. Sequences from humans, specifically those not included in the samples used for constructing the human reference, may align incorrectly or not at all when they originate from a region that differs from the reference genome. The reliance on a single reference human genome could introduce substantial biases in downstream analyses and result in an inability to detect disease-related genetic variants.

A series of large-scale projects have characterized >660 million single-nucleotide polymorphisms (SNPs; in dbSNP[10]) and >10 million structural variants (in dbVar[11]). Although these variants represent a valuable resource for genetic analyses, existing computational tools do not adequately incorporate them. To address these challenges, we present a genome indexing scheme that uses a graph-based approach to capture a wide representation of genetic variants with very low memory requirements. More than a decade ago, adaptation of the Burrows–Wheeler transform (BWT) and Ferragina Manzini (FM) index[12,13] in read alignment programs such as Bowtie[14], BWA[15] and SOAP2 (ref. [16]) enabled alignment that was two to three orders of magnitude faster than with alignment programs such as BLAT[17] and MAQ[18], with similarly low memory requirements. We built a new alignment system, HISAT2, that enables a fast search through its graph index. Unlike other graph aligner algorithms that use memory-intensive *k*-mer-based indexes, such as vg[19] and bpa aligner[20], we implemented a graph FM (GFM) index that makes HISAT2 the most practical and efficient method for aligning sequencing reads to a graph that captures the entire human genome along with a large number of variants.

Our graph-based alignment approach enables much higher alignment sensitivity and accuracy than standard, 'linear' reference-based alignment approaches, especially for highly polymorphic genomic regions. Representing and searching through the numerous alleles of even one gene has long been a challenge, requiring a large amount of compute time and memory. For example, the *HLA-A* gene, for which alleles must be matched precisely between donors and recipients of organs and stem cell transplants, has more than 3,000 identified alleles. Computational methods have so far focused on genotyping only one or a few genes because whole-genome genotyping has simply been impractical. Using HISAT2 as a foundation, we developed HISAT-genotype to compute the HLA type and the DNA 'fingerprint' of a human using standard whole-genome sequencing data. Because HISAT-genotype works well for multiple highly diverse genes and genomic regions, we expect it will be straightforward to extend it to many more known variants in human genes.

Results

HISAT2 and HISAT-genotype algorithms. HISAT2 implements a graph-based data structure and an alignment algorithm to enable fast and sensitive alignment of sequencing reads to a genome and a large collection of small variants. HISAT2 also implements an

[1]Lyda Hill Department of Bioinformatics, University of Texas Southwestern Medical Center, Dallas, TX, USA. [2]Department of Computer Science, Stanford University, Stanford, CA, USA. [3]Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, School of Medicine, Johns Hopkins University, Baltimore, MD, USA. [4]Departments of Biomedical Engineering, Computer Science, and Biostatistics, Johns Hopkins University, Baltimore, MD, USA. *e-mail: daehwan.kim@utsouthwestern.edu

indexing algorithm for repeat sequences in a genome in which alignments of a repetitive read are projected to one location and later fully recovered. HISAT-genotype uses HISAT2 as an alignment engine along with additional algorithms to carry out HLA typing and DNA fingerprinting analysis.

HISAT2 begins by creating a linear graph of the reference genome and then adds mutations, deletions and insertions as alternative paths through the graph. Figure 1a illustrates how variants are incorporated using a short reference sequence, GAGCTG. In the graph representation, bases are represented as nodes and their relationships are represented as edges. The figure shows three variants: a SNP where T replaces A, a deletion of a T and an insertion of an A. Although the example shows only single-base polymorphisms, HISAT2 can incorporate insertions of up to 20 base pairs (bp) and deletions of any length.

In the genome graph data structure, any path in the graph defines a string of bases that occur in the reference genome or one of its variants. For example, the path G→A→G→C defines the string GAGC. Strings can be ordered lexicographically; for example, AGC comes before GTG, which comes before TGZ. A special symbol, Z, is used to indicate the end of the graph and to properly sort strings. To allow fast alignment of reads to the genome graph, we first converted the graph into a prefix-sorted graph using a method developed by Sirén et al.[21] This prefix-sorted graph is more efficient for search and storage. The prefix-sorted graph is equivalent to the original one in the sense that they define the same set of strings. In a prefix-sorted graph, nodes are sorted such that any strings from a node with a higher lexicographic rank appear before any strings from a node with a lower rank. For example, any string from the node ranked first (node A in Fig. 1a), such as AGCTGZ, comes before any strings from any other nodes. An equivalent table for this prefix-sorted graph is shown in Fig. 1a. The table stores two types of information. For outgoing edges, given node rankings 1 to 11, the label of each node is stored according to its number of outgoing edges. Here node rankings are also referred to as node IDs. For example, node 1 has one outgoing edge, from A to G, so this node's label A is stored once, as shown in the first row under 'First' of the 'Outgoing edge(s)' columns. Node 3 has three outgoing edges, so this node's label C is stored three times. For incoming edges, given the node rankings, the labels of the preceding nodes are stored. For example, node 1 has one incoming edge from the node labeled G, so this label G is stored once in the first row under 'Last' of the 'Incoming edge(s)' columns. Node 5 has two incoming edges from nodes labeled A and T, so A and T are stored accordingly.

Although edges are not directly stored using node IDs (Fig. 1a), we can implicitly construct the edge information using a key property of the table representation, called last-first (LF) mapping. The LF mapping property says that the *i*th occurrence of a certain label in the last column corresponds to the *i*th occurrence of that label in the first column. For example, node 3 has an incoming edge from the node labeled G. This is the second occurrence of G in the last column of the table, which corresponds to node 5 in the first column, as shown with two blue arrows connected by a dashed line in Fig. 1a. This indirect representation of edges leads to a substantial reduction in memory requirements for storing the table. The table representation can be further compacted using the scheme illustrated in Supplementary Fig. 1.

To further improve speed and accuracy, we modified the hierarchical indexing scheme from HISAT[22] to create a hierarchical GFM (HGFM) index. In addition to the global index for representing the human genome plus a large collection of variants, we built thousands of small indexes, each spanning ~57 kb, which collectively cover the reference genome and its variants (Fig. 1b). This approach provides two main advantages: (1) it allows a search on a local genomic region (57,344 bp), which is particularly useful for aligning RNA-seq reads spanning multiple exons, and (2) it provides a much faster lookup as compared to a search using the larger global index, owing to the local index's small size. In particular, these local indexes are so small that they can fit within the cache memory of a CPU's cache memory, which is substantially faster than standard RAM.

Our implementation of this new scheme uses just 6.2 GB for an index that represents the entire human genome plus ~14.5 million common small variants, which include ~1.5 million insertions and deletions available from dbSNP (v.144). The incorporation of these variants requires only 50–60% more CPU time than HISAT2 (among the fastest alignment programs) searching the human genome without variants and it obtains greater alignment accuracy for reads containing SNPs (Supplementary Tables 1–4). Additional details about sequence search via graph index and about the algorithms to handle mismatches and indels are given in the Methods and our earlier work on HISAT[22].

Indexing repeat sequences (HISAT2). On the basis of sets of 100-bp simulated and 101-bp real reads that were used in our evaluation (Supplementary Note 1), we found that 2.6–3.4% and 1.4–1.8% of the reads were mapped to ≥5 locations and ≥100 locations, respectively. For such reads, commonly used alignment programs report only one or a few randomly chosen locations. Even if a program could report all alignments, attempting to do so would likely consume a prohibitive amount of disk space. To address this issue, we developed a new indexing and alignment strategy that combined a set of identical sequences from the reference genome into one representative sequence that we called a repeat sequence, and directly aligned reads to that repeat sequence, resulting in one repeat alignment per read (Methods; Fig. 1b).

HISAT2 has an option to report repeat alignments (Fig. 1b). If a read matches a repeat sequence, the read is aligned to just one location (the repeat sequence) instead of being aligned to the corresponding real locations of the genome. This dramatically decreases the number of alignments that must be reported. For example, in one of our simulated read sets (10 million 100-bp reads), the total number of alignments was 108,698,299 when all alignments were reported. When we combined alignments to identical sequences in the reference genome, the total number of alignments decreased to 10,618,348 and the alignment file size (in SAM format) decreased from 29.5 GB to 3 GB.

Identifying gene sequences and genomic regions with HISAT-genotype. Building on the HISAT2 graph representation, we then set out to create an algorithm to perform genotyping from a shotgun sequencing dataset, focusing initially on two distinct applications of genotype: (1) the human HLA region, a highly variable region that is used to determine compatibility between donors and recipients in organ transplants, and (2) DNA fingerprinting, in which 13 specific regions are tested to determine whether a DNA sample matches a particular subject.

There is currently no centralized database for the many known genomic variants in human populations. Instead, each database has its own data format and naming conventions. To address this challenge, we parsed exterior databases (for example IMGT/HLA[23] and CODIS[24]) for human genes or genomic regions and converted them into an intermediate format upon which several HISAT-genotype algorithms are conveniently built. We created a graph genome, called a 'genotype genome', which is specifically designed to aid in carrying out genotyping (Fig. 2). In addition to variants and haplotypes, the genotype genome includes additional sequences inside the consensus sequence shown in yellow, resulting in substantial differences in coordinates with respect to the human reference genome. Thus, a genotype genome should not be used for purposes other than genotyping analysis.

In contrast to linear-based representations of the human reference augmented by sequences representing gene alleles, graph

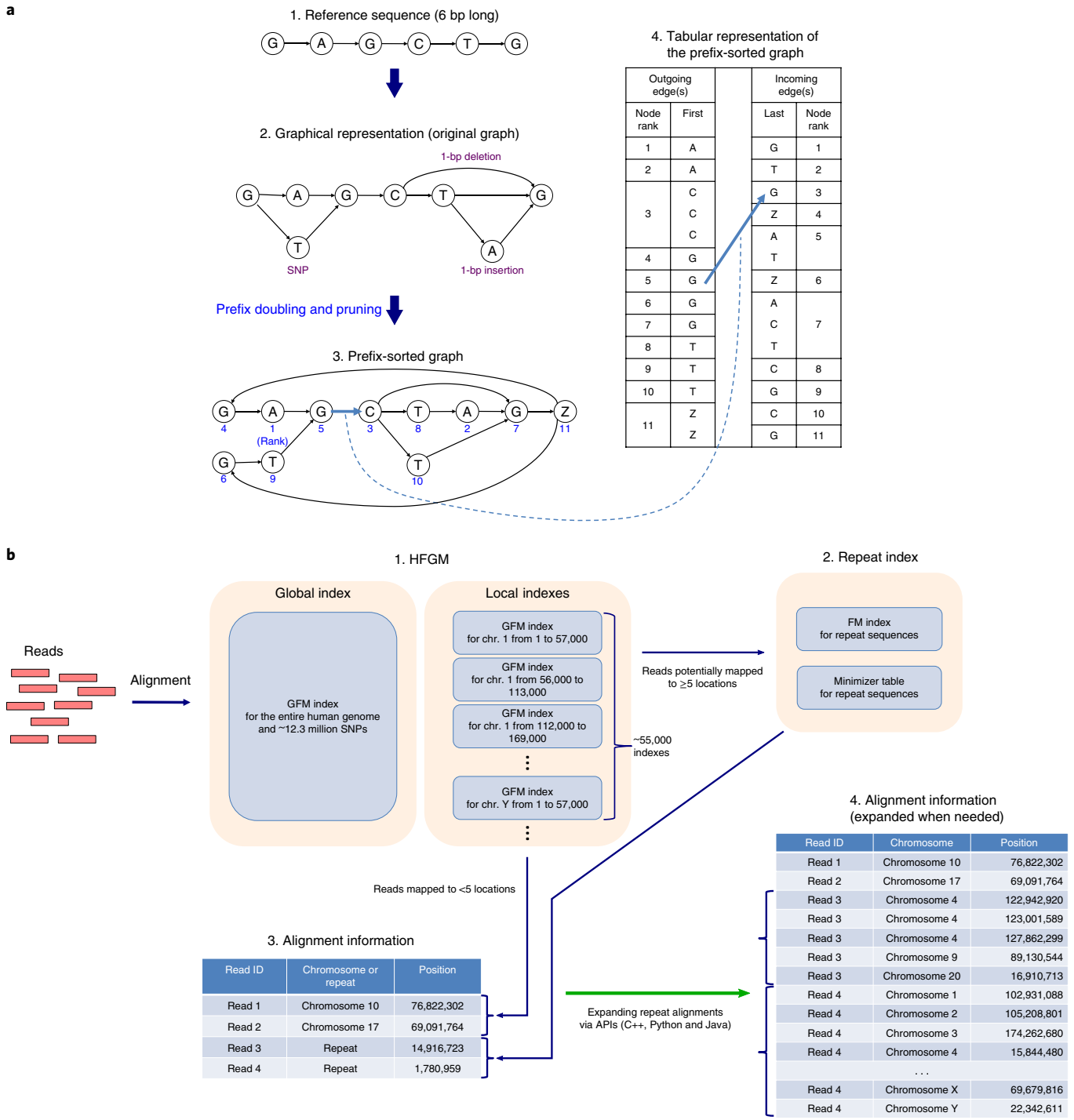

Fig. 1 | Graph representation with its tabular form and HISAT2 indexes and alignment output. a, Graph representation of indels and mutations and its tabular representation. Starting with a 6-bp reference sequence, GAGCTG (1), the lower graph (2) incorporates three variants: a single-nucleotide variant (A/T), a 1-bp deletion (T) and a 1-bp insertion (A). A prefix-sorted version of the graph (3) has 11 nodes and 14 edges. Each node has a unique numerical node ID shown in blue to indicate its lexicographic order (1 being the first) with respect to the other nodes in the graph. The node labeled with 'Z' demarcates the end of the reference sequence. The table on the right (4) has two columns under 'Outgoing edge(s)' that show the node IDs and their labels repeated according to the number of their outgoing edges (node 3, labeled C, is repeated three times with three outgoing edges to nodes 7, 8 and 10). The table has two columns under 'Incoming edge(s)' that show the node IDs and the 14 labels for the preceding nodes (G is the preceding label for node 1, A and T for node 5). The table is more compact in memory usage than the graph representation. **b**, Overview of HISAT2's indexes and alignment output. (1) Hierarchical indexing in the HGFM index. Hierarchical indexing consists of two types of indexes: (1) a global index that represents the entire human genome and (2) 55,172 overlapping local indexes that collectively cover the genome plus all variants. When both are GFM indexes, a genome plus a large collection of variants can be searched simultaneously. (2) A repeat index represents genomic sequences that are identical. (3) A read-matching repeat sequence (for example, read3 and read4) is aligned to just one location (the repeat sequence). (4) The corresponding genomic locations of repeat-aligned reads are retrieved via application programming interfaces (APIs). Chr: chromosome.

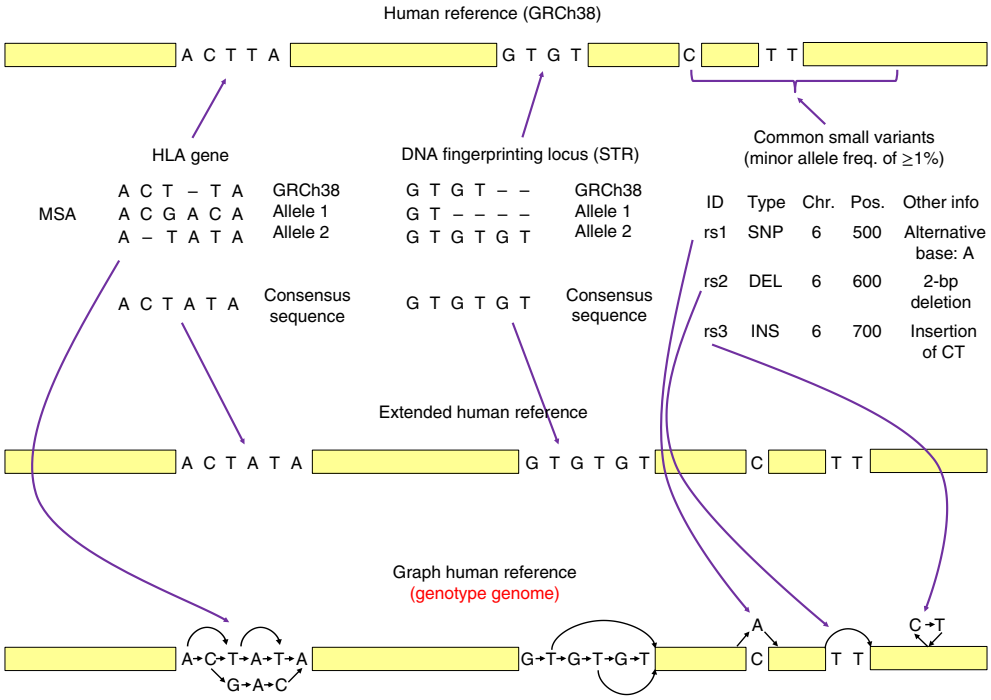

Fig. 2 | Construction of the graph human reference, that is a Genotype genome. The figure illustrates how HISAT-genotype extends the human reference genome (GRCh38) by incorporating known genomic variants from several well-studied genes, DNA fingerprinting loci and common small variants (variants with minor allele frequencies of ≥1%) from dbSNP. Top, the process begins with analyzing information found in the selected databases to construct consensus sequences. The IMGT/HLA database includes over 15,500 allele sequences for 26 HLA genes. A consensus sequence for each HLA gene is constructed on the basis of the most frequent bases that occur in each position of the multiple-sequence alignments. NIST STRBase contains allele sequences for 13 DNA fingerprinting loci. Because the sequences of the 13 loci are short tandem repeats, HISAT-genotype chooses the longest allele for each locus as a consensus sequence. Middle, the human reference is extended by replacing the HLA genes and 13 DNA fingerprinting loci with their consensus sequences. Bottom, the known genomic variants are then incorporated into the extended references using HISAT2's graph data structure. Common small variants from dbSNP such as SNPs, deletions and insertions are also incorporated into the extended reference. In HISAT-genotype this graph reference is called a genotype genome. MSA, multiple sequence alignment.

representations are much more efficient in terms of memory usage and/or alignment speed (Supplementary Fig. 2). When working with whole-genome sequencing data, using the right reference/index is crucial. Much greater alignment accuracy can be achieved by using the reference that most closely matches the genome from which reads originated. Using the wrong reference (for example just a few genes instead of the whole genome) can lead to reads being incorrectly aligned (Supplementary Fig. 3).

Once reads are extracted that belong to a particular gene or genomic region using a genotype genome, HISAT-genotype performs two further downstream analyses based on the read alignments: (1) typing and (2) gene assembly. Typing is the process of identifying the two alleles (or the one allele if an individual is homozygous) for a particular gene that best match a given sequencing dataset. When paired-end reads of ≥100 bp with a sequencing depth of at least 30–50× coverage are used, HISAT-genotype is frequently able to assemble full-length alleles and determine whether they are novel by comparing the assembled alleles to known alleles in the database, as described below.

Instead of directly assembling reads on the basis of overlaps among reads, HISAT-genotype splits aligned reads into fixed length segments called *k*-mers, as done in de Bruijn graph assemblers[25,26]. These *k*-mers form an assembly graph that enables the systematic assembly of alleles by handling noise and resolving assembly ambiguities (Fig. 3 and Supplementary Dataset 1).

Figure 3b illustrates the assembly of two distinct alleles using the *k*-mer assembly graph. HISAT-genotype assumes that each locus should have at most two alleles, which means that one of the three *k*-mers in Fig. 3b needs to be removed. HISAT-genotype uses the number of reads that support each *k*-mer to make this choice. For example, if the *k*-mers shown in green and yellow are supported by three reads each, while the *k*-mer in red is only supported by one read, the program removes the *k*-mer in red. After noise removal, it is not yet clear which *k*-mers are linked to other *k*-mers from the same allele (for example, the yellow and green nodes). Read-pair information is then used to resolve this ambiguity. Suppose there are three pairs that support CGC and CCG in green (at the top of the figure). Drawing upon this read-pair information, we can resolve the ambiguity. Read pairs are not always sufficient to separate alleles; for example, two known alleles HLA-A*01:01:01:01 and HLA-A*11:01:01:01 of NA12878 have the same ~1,200-bp sequence in the middle, while typical Illumina read pairs are separated by 600 bp or less. To fully assemble alleles, HISAT-genotype makes use of alleles in the database to combine partial alleles into full-length alleles. This approach enabled HISAT-genotype to assemble correctly all HLA-A alleles for the 'platinum' genomes used in our experiments, although this strategy can introduce a bias toward known alleles.

Because many variants, including insertions and deletions, are incorporated in the genotype genome, a read can be locally aligned in multiple ways at approximately the same location (Fig. 3a), where only one alignment is actually correct. If a program selects an incorrect alignment, this may in turn lead to choosing the wrong allele. HISAT-genotype handles such cases by choosing the most likely alignment using a statistical model and an expectation–maximization (EM) method (Methods).

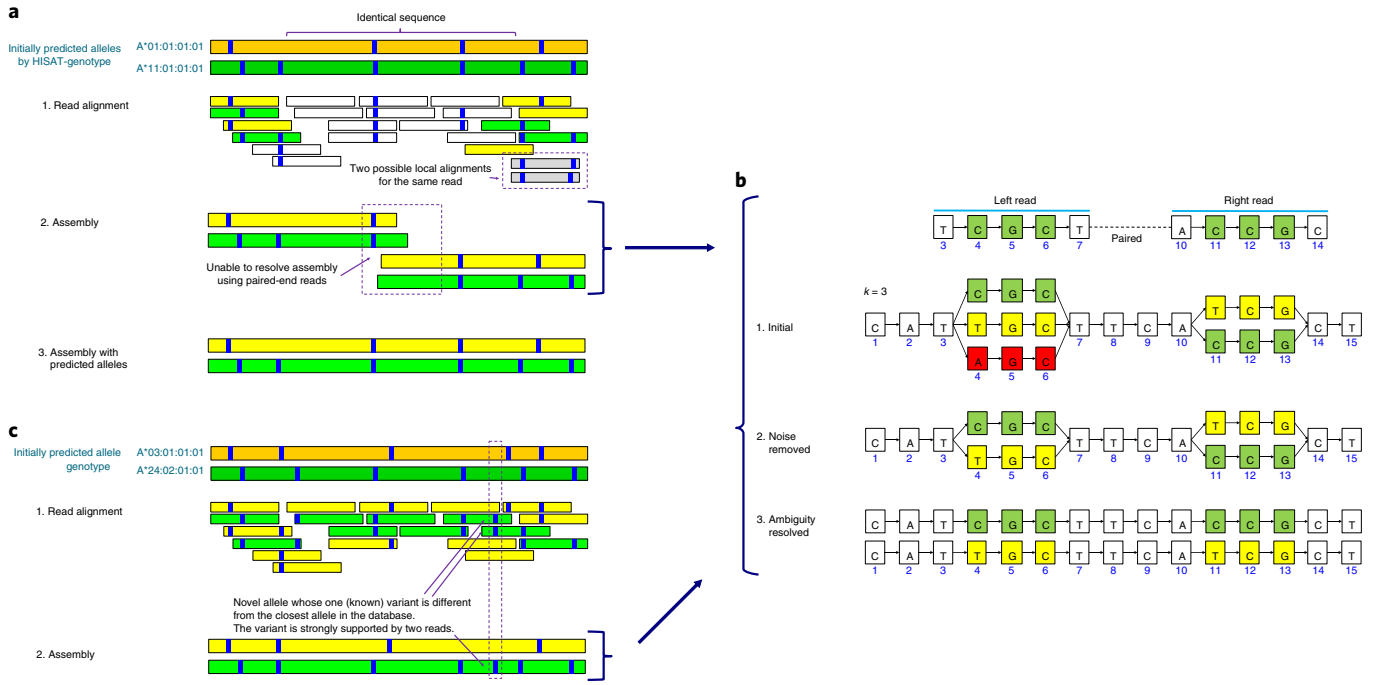

Fig. 3 | HISAT-genotype assembly of two *HLA-A* alleles through a guided *k*-mer assembly graph. a, Typing and assembly. The figure shows an abridged example of HISAT-genotype's assembly output; see Supplementary Dataset 1 for the full assembly output for NA12878. The first two bands are two alleles predicted by HISAT-genotype, in this case HLA-A*01:01:01:01 in dark green and HLA-A*11:01:01:01 in dark yellow. Each blue stripe indicates where there is a specific genomic variant with respect to the consensus sequence of the HLA-A gene. (1) Shorter bands indicate read alignments whose color is determined according to the degree of compatibility with either of the initially predicted alleles. Reads equally compatible with both alleles are shown in white. Some reads can be locally aligned, that is aligned to virtually the same location with just different variants, such as when reads are aligned with or without deletions near their ends, displayed here in gray. Because the two predicted (and in fact true/known) alleles share a large common sequence (2), read-pair information is insufficient to fully separate the alleles. **b**, HISAT-genotype splits aligned reads into fixed-length *k*-mers. In this simplified case, reads are 5 nucleotides long and $k = 3$. A pair of reads is aligned at the third location and the tenth location of the graph representation for the HLA gene. When reads have divergent *k*-mers, the graph has a corresponding number of branches. One path traversing the graph from left to right constitutes one potential allele sequence. We call this a guided *k*-mer assembly graph, with 'guided' emphasizing that *k*-mers are placed according to their aligned locations. The algorithmic details are given in the main text. (3) In addition, HISAT-genotype uses the predicted alleles to enable full-length assembly of both. **c**, Novel allele discovery: a putative novel *HLA-A* allele identified with strong computational evidence. This figure shows an abridged example of HISAT-genotype's assembly output. At the top are shown the two initially predicted alleles, which are the best matches of the data to previously known *HLA-A* alleles. The green assembled allele at the bottom, which was generated de novo by HISAT-genotype's assembler, has one variant different from the predicted allele, HLA-A*24:02:01:01. Two reads shown in green support the variant. See Supplementary Dataset 10 for a more detailed output from a similar case found in LP6005093-DNA_E03 (a CAAPA genome) at the 2,780th base.

Benchmarking and validation of HISAT2. We demonstrated HISAT2's performance on aligning sequences to the human genome, and compared it with the most widely used alignment programs, BWA-mem[27] and Bowtie2 (ref. [28]), and to vg[19], a popular graph-based alignment program (we did not use bpa aligner[20] owing to its lack of ability to handle multiply mapped reads). We did not include HISAT[22] in our evaluation because HISAT2 is a variant-aware version of HISAT with almost identical performance in terms of alignment quality, runtime, and memory usage when aligning to a linear reference. We used four sets of 20 million simulated 100-bp paired-end reads (10 million pairs) and one set of 20 million real 101-bp paired reads (10 million pairs), which are the first 20 million reads from a larger set taken from NA12878 (ref. [29]). We generated the four simulated datasets from the human reference genome (GRCh38) as follows: (1) reads including known variants with no sequencing errors; (2) reads including known variants with 0.2% per-base sequencing errors; (3) reads with no sequencing errors and no known variants (perfect reads); and (4) reads with 0.2% per-base sequencing errors and no known variants. The reads in datasets 1, 2 and 4 included up to three differences with respect to GRCh38 (Supplementary Note 1).

We ran two versions of HISAT2, HISAT2.Graph and HISAT2.Linear, which use a graph index and a linear index for alignment, respectively. We also ran vg with a graph and a linear index, vg.Graph and vg.Linear. All programs were run with two different modes: default settings that usually allow one or a few alignments to be reported and different settings that allow more alignments to be reported (we added the suffix 'sensitive' to each program's name to indicate the latter settings, for example 'Bowtie2.sensitive').

Overall, the graph-based aligners, HISAT2.Graph (both default and sensitive settings) and vg.Graph.sensitive, provided the highest alignment sensitivity (99.08–99.19% and 98.18%, respectively) on the simulated reads that included SNPs and sequencing errors (dataset 2), followed by Bowtie2.sensitive (97.68%) and HISAT2.Linear.sensitive (97.54%), BWA-mem.sensitive (97.42%), HISAT2.Linear (default settings; 96.52%), Bowtie2 (default settings; 95.99%), and BWA-mem (94.02%) (Fig. 4). HISAT2 processed 36,735 pairs of reads per second (PPS) using default settings for HISAT2.Linear. Other speeds were 24,941 PPS in HISAT2.Linear.sensitive, 28,729 PPS in HISAT2.Graph, and 21,207 PPS in HISAT2.Graph.sensitive. Bowtie2 and BWA-mem processed 5,663 to 10,917 PPS, while vg processed only 1,012 to 1,346 PPS. Bowtie2 required the smallest

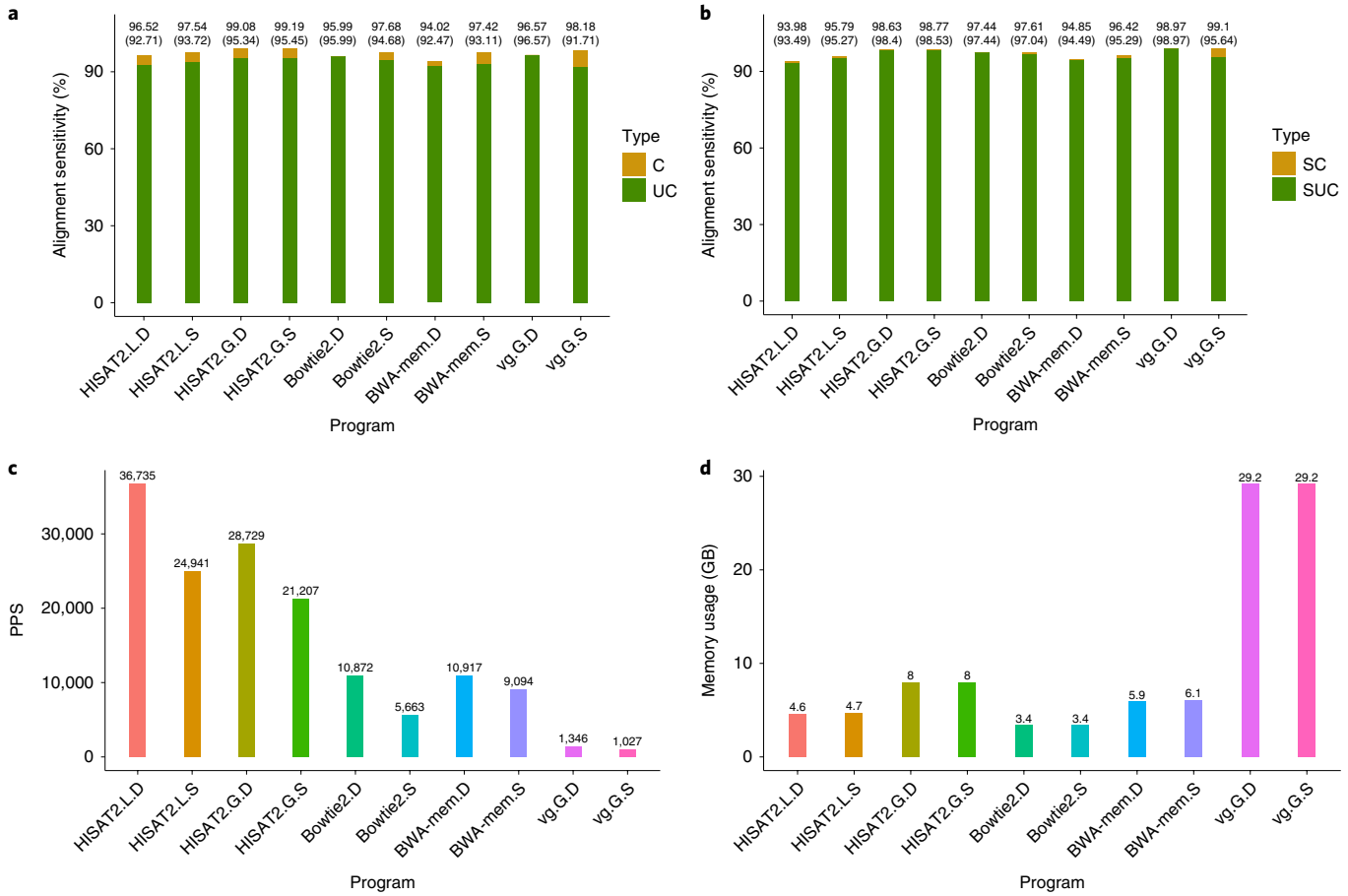

Fig. 4 | Comparisons of HISAT2, Bowtie2, BWA-mem, and vg using 10 million simulated read pairs that include SNPs. a, Alignment sensitivity for all 10 million simulated read pairs. **b**, Alignment sensitivity for only those read pairs that include SNPs. **c**, Alignment speed. **d**, Memory requirements. Alignment sensitivity is defined as the number of correctly aligned read pairs divided by the total number of read pairs. C, alignment sensitivity calculated on the basis of on any one of multiple alignments being correct. UC, alignment sensitivity calculated on the basis of on pairs being uniquely aligned. SC, alignment sensitivity similar to C, but calculated only for pairs with at least one read that includes one or more SNPs. SUC, alignment sensitivity similar to UC, but calculated only for pairs with at least one read that includes one or more SNPs. PPS, number of pairs processed per second. Parameter settings and reference types are indicated after the program names as follows: D, default alignment settings; S, sensitive alignment settings; L, linear genome alignment; G, graph genome alignment. All programs were run on the same computer, as described in Supplementary Table 5.

amount of memory (3.4 GB), followed by HISAT2.Linear (4.5 GB) and BWA-mem (5.7–6.2 GB). Graph-based aligners (HISAT2.Graph and vg) required more RAM, with HISAT2.Graph requiring slightly more memory (8 GB) than the linear-based aligners, and vg requiring 29 GB.

Programs did not perform differently between datasets 1 and 2. On reads that did not include SNPs (datasets 3 and 4), all programs with sensitive settings provided relatively high alignment sensitivity (Supplementary Table 2). For example, HISAT2, BWA-mem.sensitive, Bowtie2.sensitive, and vg.Linear.sensitive aligned the highest number of pairs (98.70–99.99% on dataset 3 and 98.61–99.81% on dataset 4). The results for dataset 3 (perfect reads) demonstrate that HISAT2 correctly mapped almost all read pairs (99.99%), including those that were mapped to ≥500 locations, while the second best program, BWA-mem.sensitive, correctly aligned 99.46% of the pairs.

The incorporation of known variants into its index enabled HISAT2.Graph to align reads 2–3 times faster than Bowtie2 and BWA-mem, which had to separately deal with mismatches due to sequence variation using time-consuming alignment algorithms (for example dynamic programming and seed chaining). Although the simulated reads only differed from the reference genome by a maximum of three edits, HISAT2.Graph was even more effective at aligning reads that included many known variants, as illustrated in our HLA and DNA fingerprinting analysis below, while linear genome-based aligners may have had difficulty aligning such divergent reads.

Supplementary Table 3 shows the results of all the aligners using a real set of paired-end reads. Because we do not know the true alignments for these reads, we evaluated performance using the accumulated alignment ratio with edit distance (0–6) in both reads of a pair, pairs processed per second, and memory requirements. The overall alignment ratios were similar among the programs, ranging from 92.3–93.1%. HISAT2.Graph and vg.Graph had a higher number of pairs aligned at small (0–2) edit distances. For example, HISAT2.Graph and vg.Graph (default settings) aligned 78.7% and 78.0% of pairs perfectly (for example, zero edit distance), while others aligned 67.0–67.6%. This is mainly because HISAT2.Graph does not impose an edit distance 'penalty' for mismatches due to known SNPs while others impose a penalty.

HISAT-genotype for HLA genotyping and DNA fingerprinting. To demonstrate applications of HISAT2, we describe genotyping the HLA genes (*HLA-A*, *HLA-B*, *HLA-C*, *HLA-DQA1*, *HLA-DQB1* and *HLA-DRB1*) and evaluating DNA fingerprinting loci using 13 markers plus the sex-determining marker gene amelogenin

Table 1 | HISAT-genotype DNA fingerprinting results

	Genome ID																
	NA12877	**NA12878**	**NA12879**	**NA12880**	**NA12881**	**NA12882**	**NA12883**	**NA12884**	**NA12885**	**NA12886**	**NA12887**	**NA12888**	**NA12889**	**NA12890**	**NA12891**	**NA12892**	**NA12893**
HISAT-genotype																	
AMEL	X, Y	X	X	X	X	X, Y	X, Y	X, Y	X	X, Y	X	X, Y	X, Y	X	X, Y	X	X, Y
CSF1PO	10	10, 11	10	10	10	10	10, 11	10, 11	10	10	10	10, 11	10, 11	10, 13	9, 11	10, 12	10
D5S818	10, 11	12	11, 12	11, 12	11, 12	11, 12	10, 12	11, 12	10, 12	**10, 12**	10, 12	11, 12	10, 11	11, 12	12, 13	11, 12	11, 12
D7S820	10, 13	8, 10	10, 13	10	8, 13	8, 13	8, 13	8, 10	10, 13	**10, 13**	10, 13	8, 13	10	12, 13	8, 12	10, 12	10, 13
D13S317	13	11, 12′	11, 13	11, 13	12′, 13	11, 13	11, 13	12′, 13	12′, 13	12′, 13	**11, 13**	11, 13	8, 13	11′, 13	8, 11	9, 12′	11, 13
D16S539	11	10, 11	11	11	10, 11	10, 11	10, 11	10, 11	11	10, 11	10, 11	10, 11	11, 12	11, 13	10, 13	11, 12	10, 11
D21S11	28, 29	30, 30′	29, 30	28, 30′	29, 30′	28, 30	29, 30	29, 30′	28, 30′	28, 30	28, 30′	28, 30′	28, 29	29, 32.2	25.2′, 30	30′, 32.2	**28, 30′**
TH01	6, 10	7, 9.3	6, 7	7, 10	6, 7	9.3,10	7, 10	7, 10	7, 10	**7, 10**	6, 9.3	7, 10	7, 10	6, 7	6, 9.3	7, 9.3	6, 7
TPOX	9, 11	8	8, 11	8, 11	8, 11	8, 11	8, 11	8, 9	8, 9	8, 9	8, 9	8, 11	8, 9	8, 11	8	8, 9	8, 11
vWA	16, 19	15, 17	15, 16	16, 17	17, 19	17, 19	17, 19	16, 17	15, 16	17, 19	15, 19	15, 19	18, 19	14, 16	17	15, 18	**15, 19**
D3S1358	16, 16′	16′, 17′	16, 17′	16, 16′	16′,17′	16′	16, 17′	16, 17′	16, 16′	16, 16′	16′	16, 16′	16′	14, 16	16, 16′	17′	**16, 17′**
D8S1179	13′, 14′	12	12, 13′	12, 13′	12, 14′	12, 13′	12, 13′	12, 13′	12, 13′	12, 14′	12, 14′	12, 13′	13, 14′	13, 13′	12, 15′	10, 12	12, 13′
D18S51	12, 15	16, 17	12, 17	12, 17	12, 17	12, 16	12, 17	12, 17	12, 17	12, 17	15, 17	12, 17	15, 17	12, 15	14, 17	14, 16	**12, 16**
FGA	24, 25	22, 24	24, 25	22, 24	22, 25	22, 24	24, 25	22, 24	24	22, 24	24	24, 25	20, 24	20, 25	22, 23	18, 24	24, 25
Wet-lab (Promega Fusion24)																	
AMEL	X, Y	X	X	X	X	X, Y	X, Y	X, Y	X	X, Y	X	X, Y	X, Y	X	X, Y	X	X, Y
CSF1PO	10	10, 11	10	10	10	10	10, 11	10, 11	10	10	10	10, 11	10, 11	10, 13	9, 11	10, 12	10
D5S818	10, 11	12	11, 12	11, 12	11, 12	11, 12	10, 12	11, 12	10, 12	**9, 10, 12**	10, 12	11, 12	10, 11	11, 12	12, 13	11, 12	11, 12
D7S820	10, 13	8, 10	10, 13	10	8, 13	8, 13	8, 13	8, 10	10, 13	**7, 10, 13**	10, 13	8, 13	10	12, 13	8, 12	10, 12	10, 13
D13S317	13	11, 12	11, 13	11, 13	12, 13	11, 13	11, 13	12, 13	12, 13	12, 13	**11, 12, 13**	11, 13	8, 13	11, 13	8, 11	9, 12	11, 13
D16S539	11	10, 11	11	11	10, 11	10, 11	10, 11	10, 11	11	10, 11	10, 11	10, 11	11, 12	11, 13	10, 13	11, 12	10, 11
D21S11	28, 29	30	29, 30	28, 30	29, 30	28, 30	29, 30	29, 30	28, 30	28, 30	28, 30	28, 30	28, 29	29, 32.2	25.2, 30	30, 32.2	**28.1, 30**
TH01	6, 10	7, 9.3	6, 7	7, 10	6, 7	9.3,10	7, 10	7, 10	7, 10	**7, 10, 10.3**	6, 9.3	7, 10	7, 10	6, 7	6, 9.3	7, 9.3	6, 7
TPOX	9, 11	8	8, 11	8, 11	8, 11	8, 11	8, 11	8, 9	8, 9	8, 9	8, 9	8, 11	8, 9	8, 11	8	8, 9	8, 11
vWA	16, 19	15, 17	15, 16	16, 17	17, 19	17, 19	17, 19	16, 17	15, 16	17, 19	15, 19	15, 19	18, 19	14, 16	17	15, 18	**19**
D3S1358	16	16, 17	16, 17	16	16, 17	16	16, 17	16, 17	16	16	16	16	16	14, 16	16	17	
D8S1179	13, 14	12	12, 13	12, 13	12, 14	12, 13	12, 13	12, 13	12, 13	12, 14	12, 14	12, 13	13, 14	13	12, 15	10, 12	12, 13
D18S51	12, 15	16, 17	12, 17	12, 17	12, 17	12, 16	12, 17	12, 17	12, 17	12, 17	15, 17	12, 17	15, 17	12, 15	14, 17	14, 16	**16**
FGA	24, 25	22, 24	24, 25	22, 24	22, 25	22, 24	24, 25	22, 24	24	22, 24	24	24, 25	20, 24	20, 25	22, 23	18, 24	24, 25

Eight cases in which HISAT-genotype and the wet-lab results disagree, yet HISAT-genotype appears to have identified the alleles correctly, are indicated in bold. These are explained in detail in the main text.

(*AMELX* and *AMELY*). We selected HLA genes because they are among the most diverse human genes, and selected DNA fingerprinting loci because they are short-tandem-repeat regions considerably differing in length among individuals. Algorithms to perform these two genotyping assays were implemented in the HISAT-genotype program (Methods).

HLA typing for a family of 17 genomes. The IMGT/HLA database[23] encompasses >16,000 alleles of the HLA gene family. We built a HISAT2 index of the human genome that incorporated all of these variants, which increased the computational resource requirements only slightly as compared to an index without the variants. For highly polymorphic regions such as those containing the HLA genes, HISAT2 was more sensitive than other short-read aligners; for example, on one of our datasets, HISAT2 mapped up to twice as many reads to the HLA genes as Bowtie2 (ref. [28]) (Supplementary Table 4).

The HLA allele nomenclature uses a set of four numbers to designate, from left to right, alleles classified by (1) allele group according to serological and cellular specificities and then further subgrouped by (2) protein sequence, and similarly subcategorized according to (3) coding and then (4) noncoding sequences; for example, HLA-A*01:01:01:01 is a specifier for one allele of the *HLA-A* gene. HISAT-genotype reports alleles for all four fields, unlike many other programs, which tend to report a subset of the numbers (typically the first two numbers). We conducted computational experiments using Illumina's Platinum Genomes (PG), which consist of 17 genomes (CEPH pedigree 1463; Supplementary Fig. 4) that have been sequenced previously (whole-genome sequencing data are available[29], hereafter referred to as PG data). Alleles of *HLA-A*, *HLA-B* and *HLA-C* for the NA12878, NA12891, and NA12982 genomes have been identified previously using targeted sequencing[30]. A recent study[31] reported the alleles of all six HLA genes for the 17 genomes by applying several computational methods to the PG data, with the results corresponding to the pedigree. Our experiments show that HISAT-genotype's results exactly match known alleles and computationally identified alleles of the six genes for the 17 genomes. HISAT-genotype's speed surpasses that of other currently available methods, primarily owing to HISAT-genotype's alignment engine, HISAT2 (Supplementary Table 6 and Supplementary Dataset 2).

In addition to identifying alleles for each genome, HISAT-genotype can use raw whole-genome sequence data to assemble and report full-length sequences for both alleles of each of the six HLA genes, including exons and introns (Supplementary Dataset 3 shows the full assembly output for the *HLA-A* genes of NA12892). The complete sequences of *HLA-A* reported by HISAT-genotype on the 17 genomes were all in perfect agreement with those previously reported. The assembled sequences for *HLA-B*, *HLA-C*, *HLA-DQA1* and *HLA-DQB1* were nearly identical to the previously reported ones. The sequences assembled for *HLA-DRB1* were accurate but somewhat fragmented, consisting of a small number of contigs. Greater read lengths should enable HISAT-genotype to produce complete sequences for the *HLA-DRB1* gene. Additional population-scale (917 genomes) experimental results, which included the potential discovery of new HLA alleles, and comparison with Kourami[31], another recently released method, are described in Supplementary Note 1 (see also Supplementary Tables 7 and 8, and Supplementary Datasets 4 and 5).

DNA fingerprinting. DNA fingerprinting analysis has been widely used in criminal investigations and paternity testing since its introduction in the mid-1980s. It considers a set of 13 highly polymorphic regions that in combination can identify individuals or their close relatives. The billions of reads in a whole-genome sequencing run include those from the 13 genomic regions used for DNA fingerprinting analysis. In addition to running HISAT-genotype on the whole-genome sequencing data, we performed traditional wet-lab-based DNA fingerprinting using DNA samples of the 17 PG genomes (Epstein–Barr virus-transformed B lymphocytes), which were purchased from the Coriell Institute, and a DNA fingerprinting kit (PowerPlex Fusion System; Promega).

HISAT-genotype's initial results for the PG data almost perfectly matched our wet-lab results for 11 of 13 DNA fingerprinting loci on all 17 genomes and correctly determined sex (using the amelogenin locus) for all 17 genomes (Supplementary Datasets 6 and 7). To identify the potential sources of the discrepancies for the loci that were not in perfect agreement, we examined the raw PG sequencing data and found that the NIST database used by HISAT-genotype (Supplementary Dataset 8) was missing some alleles of the 17 PG genomes (Supplementary Dataset 9). After incorporating the missing alleles, HISAT-genotype's results perfectly matched the wet-lab results for all but eight cases, which are indicated in bold in Table 1.

Assuming there were no germline or somatic mutations in the PG cell lines, an analysis of the eight disagreements indicated that HISAT-genotype was correct in all eight cases. For example, on genome NA12886 at locus D5S818, HISAT-genotype reported two alleles 10 and 12, and the wet-lab method reported three alleles 9, 10, and 12. The pedigree information (Supplementary Fig. 4) shows that NA12886's father (NA12877) had two alleles 10 and 11, and the mother (NA12878) was homozygous for allele 12, suggesting that allele 9 detected by the wet-lab method was likely a false positive. Another example is NA12877's D3S1358 locus, for which HISAT-genotype gave more specific results that consisted of two different alleles 16 and 16′, which were of the same length but were slightly different in their sequences (allele 16: TCAT followed by 3 repeats of TCTG, then followed by 12 repeats of TCTA; and allele 16′: TCAT followed by 2 repeats of TCTG, then followed by 13 repeats of TCTA). Because the two alleles had identical lengths, the wet-lab method could not distinguish them and reported just one allele. The eight novel alleles from the 17 PG datasets in eight cases where the HISAT-genotype and the wet-lab results disagreed yet HISAT-genotype seemed to have identified correctly are indicated in bold.

Discussion

Our original implementation of the GFM index in HISAT2 aligned reads to the human genome using just 7.6 GB of memory (RAM), an amount available on most desktop and laptop computers, in contrast to the 20 GB of RAM (or more) required by other graph aligners. Through algorithmic innovations, HISAT2 processed reads at a speed comparable to the widely used linear aligners Bowtie2 and BWA-mem. HISAT2 can align whole-genome, exome, or transcriptome sequencing reads produced by Illumina sequencers. By default, the system allows up to three mismatches (or, similarly, an edit distance of up to 3) per read, in addition to any number of known SNPs, which do not count as mismatches.

With the graph representation and the search capability made feasible by HISAT2 and other graph aligners, one might contemplate incorporating all known variants, including rare ones, into an all-in-one pan-genome graph index. Although this idea is appealing, it would likely prove ineffective because reads would likely map to multiple and often incorrect locations, and in addition this approach might create performance issues such as slower runtimes and high memory requirements. Instead, a graph index containing common small variants, as we describe here, may be more practical. We note that the current HISAT2 index covers 92% of known variants of the NA12878 genome and is likely to cover a similar percentage of other human genomes as variant databases (for example dbSNP) expand. Instead of one massive pan-genome graph representation, having dozens or hundreds of reference genomes combined with sets of relevant variants using graph representations may be a more appropriate strategy. We found that HISAT2 with its

graph genome of common SNPs was able to identify more known SNPs of the NA12878 genome (99.4%) than Bowtie2 and BWA (98.9–99.1%) (Supplementary Note 1 and Supplementary Table 9).

We have demonstrated the effectiveness of HISAT-genotype for typing and assembling HLA genes. Future versions of HISAT-genotype will be extended to enable typing and phasing of all regions in the human genome, with the goal of producing a fully phased individual genome.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of code and data availability and associated accession codes are available at https://doi.org/10.1038/s41587-019-0201-4.

Received: 18 April 2018; Accepted: 27 June 2019;
Published online: 2 August 2019

References

1. 1000 Genomes Project Consortium A map of human genome variation from population-scale sequencing. *Nature* **467**, 1061–1073 (2010).
2. 1000 Genomes Project Consortium An integrated map of genetic variation from 1,092 human genomes. *Nature* **491**, 56–65 (2012).
3. GTEx Consortium The Genotype-Tissue Expression (GTEx) project. *Nat. Genet.* **45**, 580–585 (2013).
4. Lappalainen, T. et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501**, 506–511 (2013).
5. t Hoen, P. A. et al. Reproducibility of high-throughput mRNA and small RNA sequencing across laboratories. *Nat. Biotechnol.* **31**, 1015–1022 (2013).
6. Sanders, S. J. et al. De novo mutations revealed by whole-exome sequencing are strongly associated with autism. *Nature* **485**, 237–241 (2012).
7. Krumm, N. et al. Excess of rare, inherited truncating mutations in autism. *Nat. Genet.* **47**, 582–588 (2015).
8. Leinonen, R., Sugawara, H., Shumway, M. & International Nucleotide Sequence Database Consortium. The sequence read archive. *Nucleic Acids Res.* **39**, D19–D21 (2011).
9. Lander, E. S. et al. Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).
10. Sherry, S. T. et al. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.* **29**, 308–311 (2001).
11. Lappalainen, I. et al. DbVar and DGVa: public archives for genomic structural variation. *Nucleic Acids Res.* **41**, D936–D941 (2013).
12. Burrows, M. & Wheeler, D. J. A block sorting lossless data compression algorithm. *SRC Research Report 124* (Digital Equipment Corporation, 1994).
13. Ferragina, P. & Manzini, G. in *Proceedings 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society* 390–398 (2000).
14. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**, R25 (2009).
15. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
16. Li, R. et al. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* **25**, 1966–1967 (2009).
17. Kent, W. J. BLAT—the BLAST-like alignment tool. *Genome Res.* **12**, 656–664 (2002).
18. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **18**, 1851–1858 (2008).
19. Garrison, E. et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* **36**, 875–879 (2018).
20. Rakocevic, G. et al. Fast and accurate genomic analyses using genome graphs. *Nat. Genet.* **51**, 354–362 (2019).
21. Siren, J., Valimaki, N. & Makinen, V. Indexing graphs for path queries with applications in genome research. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **11**, 375–388 (2014).
22. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods* **12**, 357–360 (2015).
23. Robinson, J. et al. The IPD and IMGT/HLA database: allele variant databases. *Nucleic Acids Res.* **43**, D423–D431 (2015).
24. Hares, D. R. Expanding the CODIS core loci in the United States. *Forensic Sci. Int. Genet.* **6**, e52–e54 (2012).
25. Luo, R. et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience* **1**, 18 (2012).
26. Compeau, P. E., Pevzner, P. A. & Tesler, G. How to apply de Bruijn graphs to genome assembly. *Nat. Biotechnol.* **29**, 987–991 (2011).
27. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Preprint at https://arxiv.org/abs/1303.3997 (2013).
28. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie2. *Nat. Methods* **9**, 357–359 (2012).
29. Eberle, M. A. et al. A reference data set 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).
30. Erlich, R. L. et al. Next-generation sequencing for HLA typing of class I loci. *BMC Genomics* **12**, 42 (2011).
31. Lee, H. & Kingsford, C. Kourami: graph-guided assembly for novel human leukocyte antigen allele discovery. *Genome Biol.* **19**, 16 (2018).

Acknowledgements

We would like to express our thanks to K. Barnes and M. Daya for sharing Omixon's HLA results with us. We would like to thank B. Langmead and J. Pritt for their invaluable contributions to our discussions on HISAT2. We also greatly appreciate the generosity of G. Danuser and D. Reed in providing wet-lab bench space and equipment for us. This work was supported in part by the National Human Genome Research Institute under grants R01-HG006102 and R01-HG006677 to S.L.S. and by the Cancer Prevention Research Institute of Texas under grant RR170068 to D.K. All authors read and approved the final manuscript.

Author contributions

D.K. and S.L.S. performed the analysis and discussed the results of HISAT2 and HISAT-genotype. D.K. designed and implemented HISAT2 and HISAT-genotype. J.M.P. optimized the index-building algorithm of HISAT2. D.K. and C.P. implemented the repeat-indexing algorithm of HISAT2. D.K., C.P. and C.B. performed the evaluations of the various programs. D.K. performed the wet-lab experiments. D.K., C.B. and S.L.S. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41587-019-0201-4.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to D.K.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

GFM index and sequence search through the index. To perform the LF mapping, the number of times that a 'last' column label of a given row *r* occurs up to and including *r* needs to be identified, which involves counting occurrences from the top of the table down to row *r*. This counting would be prohibitively time-consuming for the 3-Gb human genome. To accelerate the process, the table is partitioned into small blocks of only a few hundred rows each. Additional numbers are stored within each block recording the number of occurrences of a specific base that appear up to that block. We also optimized the local counting process, where we counted the number of times a specific base appeared within that block. This overall indexing scheme is called a GFM index (Supplementary Fig. 5). Supplementary Figure 6 illustrates how a read that contains a known one-base insertion was aligned to the genome using a GFM.

Indexing repeat sequences (HISAT2). Given a read length *R* (for example, 100 bp), we first build a *k*-mer table from the reference genome sequence and its reverse complement together, where *k* was set to *R* and each *k*-mer must appear at least C times (for example five times) to be included. Note that we use both strands of the genome as a read is mapped to the reference and/or its reverse complement. Although we can directly use this *k*-mer table for aligning reads of length *R*, it would require a large amount of memory to store the sequences of all *k*-mers and their corresponding genomic coordinates. To reduce the memory use, we combined *k*-mers that originated from the same regions when possible. For example, suppose that there are 1,000 identical regions 200 bp in length in a reference genome. Each region has 101 100-mers with each 100-mer present in the 1,000 regions. If we were to store all coordinates of each *k*-mer, the number of all coordinates would be 101,000. However, if we can combine *k*-mers occurring in the same region into one sequence, we simply need to store one coordinate per region; thus, the number of coordinates would drop to 1,000. In practice, real genomes have identical sequences of varying length.

Supplementary Figure 7 illustrates how to merge *k*-mers into repeat s equences, where we can use any *k* as the initial value. This approach substantially reduces the number of coordinates to store. For example, the number of 100-mers that occur more than five times in the human reference genome is 4,000,527, with the average number of coordinates corresponding to each 100-mer as 19.1. This amounts to a total of 76,446,383 coordinates that we would store using the naive approach. If we allow *k*-mers to be extended to up to a certain length (for example, 300 bp), we reduce the number of coordinates to 2,825,142. We refer to both *k*-mers and extended *k*-mers as repeat sequences. When *k*-mers are extended up to 300 bp, the number of repeat sequences is reduced from 4,000,527 to 121,793.

This strategy guarantees that a read whose sequence is present $\geq C$ times in the genome is mapped to all those locations. Similarly, a read pair in which both of its sequences are present $\geq C$ times in the genome is mapped. More specifically, a read whose sequence is present *n* times ($n \geq$ C) is mapped to only one repeat sequence. The portion of the repeat sequence matching the read exactly includes *n* coordinates. This approach works perfectly for a fixed read length, *R*, which is typical of experiments using Illumina sequencers, although reads of a length close to *R* can also be handled with slightly decreased alignment sensitivity. HISAT2 also allows for building indexes of various read length and using only one (or a few) of them on an actual run so that it requires only a small amount of additional memory.

We built a BWT/FM index and a minimizer-based *k*-mer table[32] with a window size of five and $k=31$ on these repeat sequences to enable rapid alignment of 100-bp reads with up to three mismatches.

HISAT-genotype's typing algorithm. Because allele sequences may only be partially available (for example, exons only), HISAT-genotype first identifies two alleles on the basis of the sequences commonly available for all alleles, for example exons. For example, the IMGT/HLA database includes many sequences for some key exons of HLA genes, but it contains far fewer complete sequences comprising all exons, introns, and UTRs of the genes. So far 3,644 alleles have been classified for *HLA-A*. Although all alleles of *HLA-A* have known sequences for exons 2 and 3, only 383 alleles have full-length sequences available. The sequences for the remaining 3,261 alleles include either all eight exons or a subset of them. *HLA-B* has 4,454 alleles, of which 416 have full sequences available. *HLA-C* has 3,290 alleles, with only 590 fully sequenced, *HLA-DQA1* has 76 alleles with 53 fully sequenced, *HLA-DQB1* has 978 alleles with 69 fully sequenced, and *HLA-DRB*1 has 1,972 alleles, with only 43 fully sequenced. During this step, HISAT-genotype first chooses representative alleles from groups of alleles that have the same exon sequences. Next it identifies alleles in the representative alleles that are highly likely to be present in a sequenced sample. Then the other alleles from the groups with the same exons as the representatives are selected for assessment during the next step. Second, HISAT-genotype further identifies candidate alleles on the basis of both exons and introns. HISAT-genotype applies the following statistical model in each of the two steps to find maximum likelihood estimates of abundance through an EM algorithm[33]. We previously implemented an EM solution in our centrifuge system[34], and we used a similar algorithm in HISAT-genotype, with modifications to the variable definitions as follows.

The likelihood of a particular composition of allele abundance α:

$$L(\alpha \mid C) = \prod_{i=1}^{R} \sum_{j=1}^{A} \frac{\alpha_j l_j}{\sum_k^A \alpha_k l_k} C_{ij}$$

where *R* is the number of reads, *A* is the number of alleles, α_j is the abundance of allele *j*, with a sum of 1 for all *A* alleles, l_j is the length of allele *j*, and C_{ij} is 1 if read *i* is aligned to allele *j* and 0 otherwise.

Expectation (E-step):

$$n_j = \sum_{i=1}^{R} \frac{\alpha_j C_{ij}}{\sum_{k=1}^{A} \alpha_k C_{ik}}$$

where n_j is the estimated number of reads assigned to allele *j*.

Maximization (M-step):

$$\alpha'_j = \frac{n_j / l_j}{\sum_{k=1}^{A} n_k / l_k}$$

where α'_j is the updated estimate of the abundance of allele *j*. α' is then used in the next iteration.

HISAT-genotype finds the abundances α that best reflect the given read alignments, that is, the abundances that maximize the likelihood function $L(\alpha|C)$ above by repeating the EM procedure no more than 1,000 times or until the difference between the previous and current estimates of abundances, $\sum_{j=1}^{A} \left|\alpha_j - \alpha'_j\right|$, is less than 0.0001.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The CAAPA genome data are available from dbGaP (accession phs001123.v1.p1).

Code availability

HISAT2 and HISAT-genotype are open-source software freely available at https://github.com/DaehwanKimLab/hisat2. The HISAT2 package includes programs and application programming interfaces for C++, Python and JAVA that rapidly retrieve genomic locations from repeat alignments for use in downstream analyses such as variant calling, peak calling and differential gene expression analysis.

References

32. Li, H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* **32**, 2103–2110 (2016).
33. Pachter, L. Models for transcript quantification from RNA-Seq. Preprint at https://arxiv.org/abs/1104.3889 (2011).
34. Kim, D., Song, L., Breitwieser, F. P. & Salzberg, S. L. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).

nature research

Corresponding author(s): Daehwan Kim

Last updated by author(s): 8 June 2019

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☒	☐ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ 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 all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	HISAT2 and HISAT-genotype are open source available at https://github.com/DaehwanKimLab/hisat2
Data analysis	The version of HISAT2 and HISAT-genotype is v2.2.0-beta. The version of Bowtie is 2.3.4.3. The version of BWA is 0.7.17. The version of vg is v1.13.0.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Provide your data availability statement here.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	This is not relevant as the study does not include statistical analysis
Data exclusions	No data were excluded from the study
Replication	The study has references to all datasets and software versions used to ensure reproducibility of the results of the study
Randomization	This is not relevant because the study does not include statistical analysis
Blinding	This does not apply since the study does not involve case/control comparison

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging