HHS Public Access

Author manuscript

Nat Biotechnol. Author manuscript; available in PMC 2020 November 02.

Published in final edited form as:

Nat Biotechnol. 2019 August ; 37(8): 907–915. doi: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], **Steven L. Salzberg**[3,4]

[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.

Abstract

Rapid advances in next-generation sequencing technologies have dramatically changed our ability to perform genome-scale analyses. The human reference genome used for most genomic analyses represents only a small number of individuals, limiting its usefulness for genotyping. We designed a novel method, HISAT2, for representing and searching an expanded model of the human reference genome, in which a large catalogue of known genomic variants and haplotypes is incorporated into the data structure used for searching and alignment. This strategy for representing a population of genomes, along with a fast and memory-efficient search algorithm, enables more detailed and accurate variant analyses than previous methods. We demonstrate two initial applications of HISAT2: HLA typing, a critical need in human organ transplantation, and DNA fingerprinting, widely used in forensics. These applications are part of HISAT-genotype, with performance not only surpassing earlier computational methods, but matching or exceeding the accuracy of laboratory-based assays.

Introduction

Advancements in sequencing technologies and computational methods have enabled rapid and accurate identification of genetic variants in the human population. Detailed individual

*Corresponding author: Daehwan Kim, Daehwan.Kim@UTSouthwestern.edu.

Authors' 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 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.

Code availability

HISAT2 and HISAT-genotype are open-source software freely available at https://github.com/DaehwanKimLab/hisat2.

Competing financial interests

The authors declare no competing financial interests.

genomic data along with relevant clinical and environmental information promise to help improve predictions for cancer risk, inform lifestyle choices, generate more accurate clinical diagnoses, reduce adverse drug reactions and other negative side effects of treatments, and improve patient outcomes through better-targeted therapies. Although massive sequencing projects over the past decade such as the 1,000 Genomes Project[1, 2], GTEx[3], GEUVADIS[4, 5], and the Simons Simplex Collection (SSC)[6, 7] have generated trillions of reads that are available from public archives[8], our ability to make use of these enormous data sets is still quite limited. One important limitation is that most analyses must rely on the alignment of sequencing reads against the human reference genome[9] (currently GRCh38), which does not reflect genetic diversity across individuals and populations. Sequences from other humans, particularly 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. This reliance on a single reference genome can introduce significant biases in downstream analyses, and it can miss important disease-related genetic variants if they occur in regions not present in the reference genome.

A series of large-scale projects in recent years have yielded >110 million SNPs (in dbSNP[10]) and >10 million structural variants (in dbVar[11]). Although these variants represent a valuable resource for genetic analysis, current computational tools do not adequately incorporate them. To address these challenges, we have developed a novel genome indexing scheme that uses a graph-based approach to capture a wide representation of genetic variants with very low memory requirements. Over a decade ago, adaptation of the Burrows-Wheeler Transform and Ferragina Manzini Index (BWT/FM)[12, 13] in linear reference based alignment programs such as SOAP[14], Bowtie[15], and BWA[16] has enabled two or three magnitudes faster alignment than preceding alignment programs such as BLAT[17] and MAQ[18], with similarly low memory requirements. We have built a new alignment system, HISAT2, that enables fast search through its graph index. And now, in contrast to other graph aligner development approaches that use memory demanding k-mer based indexes such as in vg[19] and bpa aligner[20], we are the first to implement a Graph FM (GFM) index, which makes HISAT2 currently the most practical method available for aligning raw 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, which must be matched precisely between donors and recipients of organ and stem cell transplants, has over 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 that it will be straightforward to extend it to many more known variants in human genes. HISAT2 and HISAT-genotype are open-source software freely available at https://

daehwankimlab.github.io/hisat2/ and https://daehwankimlab.github.io/hisat-genotype, respectively.

Algorithmic details

Here we describe the algorithms underlying HISAT2 and HISAT-genotype. HISAT2 implements a novel graph-based data structure along with an alignment algorithm to enable fast and sensitive alignment of sequencing reads to a genome and a large collection of small variants. In addition, HISAT2 implements a new indexing algorithm for repeat sequences in a genome in which alignments of a repetitive read are projected to one location and later are fully recovered. HISAT-genotype uses HISAT2 as an alignment engine along with additional algorithms to perform HLA-typing and DNA fingerprinting analysis.

Graph representation of human populations and alignment (HISAT2)

The reference human genome used by most researchers, currently version GRCh38, was assembled from data representing only a few individuals, with over 70% of the reference genome sequence coming from one person[9,21]. By its very design, the reference does not include genomic variants from the human population. Sequence alignment protocols based on this single reference genome are sometimes unable to align reads correctly, especially when the source genome is relatively distant from the reference genome[22, 23]. HISAT2 begins by creating a linear graph of the reference genome, and then adds insertions, deletions, and mutations as alternative paths through the graph. Figure 1a and b illustrate how variants are incorporated using a very 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 single nucleotide polymorphism where T replaces A, a deletion of a T, and an insertion of an A. Although the example shows only 1-base polymorphisms, HISAT2 can incorporate insertions of up to 20 bps 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; e.g., 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 queries (reads) to the genome graph, we first convert the graph into a *prefix-sorted graph* using a method developed by Sirén et al[24]. 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 Figure 1c), such as AGCTGZ, comes before any strings from any other nodes. An equivalent table for this prefix-sorted graph is shown in Figure 1d. 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 the number of outgoing edges it has. 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 3 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 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 as depicted in Figure 1d, we can implicitly construct the edge information using a very important property of the table representation, called Last-First (LF) mapping. The Last-First 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 in Figure 1d 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 that are connected by a dotted line in Figure 1. 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 Figure 1.

To further improve both speed and accuracy, we modified the hierarchical indexing scheme from HISAT[25] to create a Hierarchical Graph FM index (HGFM). 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 (Figure 2a). This approach provides two main advantages: (1) it allows search on a local genomic region (57,344 bps), which is particularly useful for aligning RNA-seq reads spanning multiple exons, and (2) it provides a much faster lookup compared to a search using the larger global index, due to the local index’s small size. In particular, these local indexes are so small that they can fit within a CPU’s cache memory, which is significantly 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 (version 144). The incorporation of these variants requires only 50~60% more CPU time compared to 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 Online methods and our earlier work on HISAT[25].

Indexing Repeat Sequences (HISAT2)

Based on sets of 100-bp simulated and 101-bp real reads that were used in our evaluation (see Results and Supplementary Note), 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. BWA-mem has a user option (-a) that enables reporting up to 500 alignments of a read. Even if a program could report all alignments, though, attempting to do so would likely consume a prohibitive amount of disk space. In order to address this issue, we have developed a novel

indexing and alignment strategy in which we combine a set of identical sequences from the reference genome into one representative sequence, which we called a repeat sequence, and directly align reads to that repeat sequence, resulting in one *repeat* alignment per read (see Online Methods for details and Figure 2b).

HISAT2 has an option to report *repeat* alignments as shown in Figure 2c and d. If a read matches a repeat sequence, then 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. The HISAT2 package includes programs and application programming interfaces (API) 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.

Identification of sequences of genes and genomic regions (HISAT-genotype)

Building on the HISAT2 graph representation, we then set out to create an algorithm to perform genotyping from a shotgun sequencing data set, focusing initially on two distinct applications of genotype: (1) the human HLA region, a highly variable region that is used to determine compatibility between donor and recipient in organ transplants, and (2) DNA fingerprinting, in which 13 specific regions are tested to determine if 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 (e.g. IMGT/HLA[26] and CODIS[27]) 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 as illustrated in Figure 3. In addition to variants and haplotypes, the genotype genome includes some additional sequences inside the consensus sequence shown in yellow, resulting in substantial differences in coordinates with respect to the human reference genome. Thus, it is important to note that 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 representations are much more efficient in terms of memory usage and/or alignment speed, as illustrated in Supplementary Figure 2. When working with whole-genome sequencing data, using the right reference/index is crucial. Much greater alignment accuracy can be achieved by using a reference that most closely matches the genome from which reads originated. Using the wrong reference (e.g. just a few genes instead of the whole genome) can lead to reads being incorrectly aligned, as depicted in Supplementary Figure 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 homozygous) for a particular gene that best match a given sequencing data set.

When paired-end reads of ≥100 bp with a sequencing depth of at least 30-50x coverage are used, HISAT-genotype is frequently able to assemble full-length alleles and determine whether they are novel by comparing the assembled alleles with known alleles in the database, as described below.

Instead of directly assembling reads based on overlaps among reads, HISAT-genotype splits aligned reads into fixed length segments called k-mers, as in done in de Bruijn graph assemblers[28,29]. These k-mers form an assembly graph (Figure 4) that enables the systematic assembly of alleles by handling noise and resolving assembly ambiguities.

Figure 4 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 Figure 4a 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 3 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 (Figure 4b), it is not yet clear which k-mers are linked to other k-mers from the same allele (e.g., 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, as shown at the top of the figure. Drawing upon this read-pair information, we can resolve the ambiguity as illustrated in Figure 4c. Read pairs are not always sufficient to separate alleles; for example, two known alleles A*01:01:01:01 and 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. In order to fully assemble alleles, HISAT-genotype makes use of alleles in the database to combine partial alleles into full-length alleles. As our results show, this approach enables 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.

Due to many variants including insertions and deletions incorporated in the *Genotype* genome, a read can be locally aligned in multiple ways at approximately the same location (Figure 4a), where only one alignment is actually correct. If a program selects an incorrect alignment, then that may in turn lead to choosing the wrong allele. HISAT-genotype handles such cases by choosing the most likely alignment using the aforementioned statistical model and EM method.

Results

Here we demonstrate HISAT2's performance on aligning sequences to the human genome, comparing it to the two most widely used alignment programs, BWA-mem[30] and Bowtie2[31], and to vg[19], the only other graph-based alignment program available. We did not include HISAT[25] 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[32]. We generated the four simulated data sets 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 data sets 1, 2, and 4 include up to 3 differences with GRCh38. For more details of the simulation, see Supplementary Note.

We ran two versions of HISAT2, HISAT2.Graph, and HISAT2.Linear, which use a graph index and a linear index for alignment, respectively. vg was also run 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, e.g. Bowtie2.sensitive).

Overall, the graph-based aligners, HISAT2.Graph (both default and sensitive settings) and vg.Graph.sensitive, provide the highest alignment sensitivity (99.08-99.19% and 98.18%, respectively) on the simulated reads that include SNPs and sequencing errors (data set 2), followed by Bowtie2.sensitive (97.68%) and BWA-mem.sensitive (97.68%), HISAT2.Linear.sensitive (97.54%), HISAT2.Linear (default settings, 96.52%), Bowtie2 (default, 95.99%), and BWA-mem (94.02%), as shown in Figure 5. 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 process 5,663 to 10,917 pps, while vg processes only 1,012 to 1,346 pps. Bowtie2 requires the smallest amount of memory (3.4 GB), followed by HISAT2.Linear (4.5 GB), and BWA-mem (5.7 to 6.2 GB). Graph-based aligners (HISAT2.Graph and vg) require more RAM, with HISAT2.Graph requiring slightly more memory (7.9 GB) than the linear-based aligners, and vg requiring 29 GB.

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

The incorporation of known variants into its index enables HISAT2.Graph to align reads 2-3 times faster than Bowtie2 and BWA-mem, which have to separately deal with mismatches due to sequence variation using time-consuming alignment algorithms (e.g. dynamic programming and seed chaining). Though the simulated reads only differ from the reference

genome by a maximum of 3 edits, HISAT2.Graph is even more effective at aligning reads that include many known variants, as illustrated in our HLA and DNA fingerprinting analysis below, while linear genome based aligners may have 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 to 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% to 93.1%. HISAT2.Graph and vg.Graph have 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 (e.g. 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 do impose a penalty.

To demonstrate applications of HISAT2, we conducted and describe the results from two experiments: (1) genotyping the human leukocyte antigen genes (HLA-A, HLA-B, HLA-C, HLA-DQA1, HLA-DQB1, and HLA-DRB1); and (2) evaluating DNA fingerprinting loci using 13 markers plus the sex-determining marker gene Amelogenin, which are markers widely used in criminal forensics to identify individuals. We selected HLA genes because they are among the most diverse human genes, and selected DNA fingerprinting loci because they are short tandem repeat (STR) regions considerably differing in length among individuals. Algorithms to perform these two genotyping assays were implemented in HISAT-genotype, as described in the Online methods section.

HLA typing for a family of 17 genomes.

The IMGT/HLA Database[26] encompasses >16,000 alleles of the HLA gene family. We built a HISAT2 index of the human genome that incorporates 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 is more sensitive than other short-read aligners; e.g., on one of our data sets, HISAT2 maps up to twice as many reads to the HLA genes as Bowtie2[31] (Supplementary Table 4).

The HLA allele nomenclature uses a set of four numbers from left to right to designate alleles first classified by (1) allele group according to serological and cellular specificities, then further sub-grouped by (2) protein sequence, and similarly subcategorized according to (3) coding and then (4) noncoding sequences; e.g., 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 consists of 17 genomes (CEPH pedigree 1463, Supplementary File 2) that have been sequenced previously (whole genome sequencing data are available[32], 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[33]. A recent study[34] 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 other currently available methods, primarily due to HISAT-genotype's alignment engine, HISAT2 (Supplementary Table 5 and Supplementary File 3).

In addition to identifying alleles for each genome, HISAT-genotype is the first method that can use raw whole-genome sequence data to assemble and report full-length sequences for both alleles of each of the 6 HLA genes, including exons and introns (Supplementary File 4 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 are all in perfect agreement with those previously reported. Its assembled sequences for HLA-B, HLA-C, HLA-DQA1, and HLA-DQB1 are nearly identical to the previously reported ones. The sequences assembled for HLA-DRB1 are 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.

HLA typing analysis at a population scale (917 genomes).

In a separate experiment, we compared HISAT-genotype with the Omixon genotyping system[35], an established commercial platform, using whole genome sequencing (WGS) data from the Consortium on Asthma among African-ancestry Populations in the Americas (CAAPA)[36] (Supplementary File 5). Table 1 shows a high concordance rate between the two methods for the allele group and protein sequences (the first two numbers of the HLA classification); more specifically, a concordance of ≥0.97 for genotyping of HLA-A, HLA-B, HLA-C, and HLA-DQA1; 0.91 for HLA-DQB1; and 0.87 for HLA-DRB1. Tests using the CAAPA data also revealed a handful of novel alleles of HLA-A and other HLA genes (Figure 6 and Supplementary File 6.).

In addition to the PG data sets, we evaluated our method using simulated data sets and compared the results of our method with those from Kourami, a recently published HLA typing method. In order to generate simulated reads, we first randomly chose 175-200 pairs of alleles each from HLA-A, HLA-B, HLA-C, HLA-DQA1, HLA-DQB1, and HLA-DRB1, resulting in a total of 1151 allele pairs. Then we generated reads from each pair of alleles that uniformly cover each allele with 20x coverage. Each allele pair was processed according to the recommended steps for HISAT-genotype and Kourami. We found that 311 of the allele pairs contained at least one allele that was not in Kourami's database. Thus, we split the data into Kourami-present alleles and all allele pairs and analyzed each, taking Kourami's ambiguous allele grouping (G groups) into consideration. As Kourami's typing uses only exonic sequences, that program groups together alleles with the same exonic sequences but different intronic sequences, then names the groups with the common prefix of the grouped alleles and a 'G' suffix. For the Kourami-present 840 allele pairs, HISAT-genotype correctly identified 99.42% at three-field resolution while Kourami identified 99.04% (Supplementary Table 6). These values change to 99.50% and 91.91% respectively when looking at all 1151 allele pairs.

HISAT-genotype and Kourami have similar calling statistics when G grouping is taken into consideration and only using alleles that are in Kourami's database. This similarity breaks down when alleles that are not in Kourami's default database are tested and/or if we do not take into consideration Kourami's ambiguous sequence groupings. This highlights an advantage in using HISAT-genotype in obtaining more direct results with no groupings and more alleles available to call by default. Additionally, HISAT-genotype can report up to four-field resolution and does so correctly 97.42% of the time when looking at all 1151 allele pairs (Supplementary File 7).

DNA fingerprinting.

DNA fingerprinting analysis has been widely used in criminal investigations and paternity testing since its introduction in the mid-1980's. 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 WGS 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 from Promega.

HISAT-genotype's initial results for the PG data almost perfectly match our wet-lab results for 11 out of 13 DNA fingerprinting loci on all 17 genomes and correctly determines sex (using the Amelogenin locus) for all 17 genomes (Supplementary File 8 and Supplementary File 9). In order 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 File 10) was missing some alleles of the 17 PG genomes (Supplementary File 11). After incorporating the missing alleles, HISAT-genotype's results perfectly match the wet-lab results for all but 8 cases, which are indicated in bold and italics in Table 2.

Assuming there are no germline and somatic mutations in the PG cell lines, an analysis of the 8 disagreements indicates that HISAT-genotype is correct in all 8 cases. For example, on genome NA12886 at locus D5S818, HISAT-genotype reports two alleles 10 and 12, and the wet-lab method reports three alleles 9, 10, and 12. The pedigree information (Supplementary File 2) shows that NA12886's father (NA12877) has two alleles 10 and 11, and the mother (NA12878) has homozygous allele 12, suggests that allele 9 detected by the wet-lab method is likely a false positive. Another example is NA12877's D3S1358 locus, for which HISAT-genotype gives more specific results that consist of two different alleles 16 and 16', which are of the same length but are slightly different in their sequences (allele 16: TCAT followed by three repeats of TCTG, then followed by twelve repeats of TCTA; and allele 16': TCAT followed by two repeats of TCTG, then followed by thirteen repeats of TCTA). Because the two alleles have identical lengths, the wet-lab method cannot distinguish them and reports just one allele.

In summary, HISAT-genotype produces highly accurate results for both HLA typing and DNA fingerprinting using whole-genome shotgun data. Compared to both targeted sequencing and wet lab methods, HISAT-genotype either matches or exceeds their

performance, sometimes discovering novel variants that could not be detected by alternative techniques.

Discussion

Our original implementation of the Graph FM index in the HISAT2 system enables the use of the smaller amount of memory typically available on a conventional desktop, compared to the 20 GB of RAM or more required by other graph aligners. Through algorithmic innovations, HISAT2 processes sequencing reads as fast as widely used linear aligners such as Bowtie2 and BWA-mem. HISAT2 uses whole genome, target captured, or transcriptome sequencing reads produced by Illumina sequencers. HISAT2 allows by default three mismatches (or similarly 3 edit distance), which can be changed to allow more differences, though this requires more runtime. We plan to expand the program to make use of long reads produced by Oxford Nanopore and PacBio sequencers, and linked reads by the 10X Genomics Chromium platform. The current version of HISAT2 allows small variants, and we plan to expand it to incorporate structure variants.

With graph representation and search capability made feasible by HISAT2 and other graph aligners, one may contemplate incorporating all known variants including rare ones into an all-in-one pan genome graph representation. Though this idea is understandably appealing, it may generate more problems than it would solve, as reads would likely map to more and often wrong locations, in addition to generating performance issues such as slow runtime and high memory requirements. Instead of creating such an exhaustive representation, a graph with common small variants may be more practical, as it covers 93% of known variants of the NA12878 genome (one of the Platinum genomes) and is expected to cover a similar percentage of other human genomes as variant databases (e.g. dbSNP) accommodate more human populations. Instead of one representation, having dozens or hundreds of reference genomes combined with sets of relevant variants using graph representations may be the more appropriate course to take. Our preliminary work shows that HISAT2 with its graph genome of common SNPs was able to identify more known SNPs (not including indels) of the NA12878 genome (99.4%) than Bowtie2 and BWA (98.9-99.1%) (see Supplementary Note). In particular, HISAT2 assigns the same alignment score whether or not reads include known variants, enabling less biased downstream analysis. HISAT2 is more capable than linear aligners when reads involve many differences with respect to the linear reference genome.

We have demonstrated the effectiveness of HISAT-genotype for typing and assembling HLA genes. HISAT-genotype will be expanded upon to enable typing and phasing of all regions in the human genome with the end goal of producing a fully phased individual genome, set of genotypes, and annotations. We plan to augment HISAT-genotype's assembly algorithm to handle more than two copies of certain genes or genomic regions (e.g. copy number variations). HISAT-genotype's output will include haplotype resolved variants in the VCF format[37] and full-length gene sequences in the FASTA format to maximize compatibility with other genomics software. The personal genome will be a valuable reference for aligning other sequencing technologies more reliably (e.g. RNA-seq) while the set of genotypes will be useful for researchers studying disease linkage.

We anticipate that with appropriate modification of HISAT2, graphs can be created for individual human's diploid genome. Given a list of variants and haplotypes for a diploid genome assembly, we first will choose a set of chromosomes as a backbone sequence and incorporate small variants from their sister chromosomes into the backbone using a graph. Structural variants such as long insertions and inversions will be appended to the backbone properly. This graph representation/index combined with a repeat index can also be effectively used for representing an individual's diploid genome. We are actively developing simple, reliable tools to convert coordinates and annotations between any human genome. These tools will enable easy query and interpretation of a personal graph genome.

Online methods

Graph FM index and sequence search through the index

In order 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 count the number of times a specific base appears within that block. This overall indexing scheme is called a Graph FM index (GFM) (Supplementary Figure 4). Supplementary Figure 5 illustrates how a query that contains a known one-base insertion is aligned to the genome using a GFM.

Indexing Repeat Sequences (HISAT2)

Given a read length R (e.g. 100-bp), we first build a k-mer table from the reference genome sequence and its reverse complement together, where k is set to R and each k-mer must appear at least C times (e.g. 5 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 combine k-mers that originate 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-bp mers with each 100-mer present in the 1,000 regions. If we were to store all coordinates of each k-mer, then the number of all coordinates would be 101,000. However, if we can combine k-mers occurring in the same region into one sequence, then 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 lengths.

Supplementary Figure 6 illustrates how to merge k-mers into repeat sequences, 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 ≥5 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 naïve approach. If we allow k-mers to be extended to up to a certain length (e.g. 300 bps), 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 bps, the number of repeat sequences is reduced from 4,000,527 to 121,793.

This strategy guarantees that a read whose sequence is present ≥C times on the genome is mapped to all those locations. Similarly, a read pair in which both of its sequences are present ≥C times on the genome is mapped. More specifically, a read whose sequence is present n times (n ≥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 lengths 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[38] with a window size of 5 and k=31 on these repeat sequences to enable rapid alignment of 100-bp reads with up to 3 mismatches.

HISAT-genotype's typing algorithm

Because allele sequences may only be partially available (e.g., exons only), HISAT-genotype first identifies two alleles based on the sequences commonly available for all alleles, e.g. 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 8 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-DRB1 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 based on 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 Expectation-Maximization (EM) algorithm[39]. We previously implemented an EM solution in our Centrifuge system[40], 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 allele j's abundance. 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 $L(\alpha \mid C)$ above by repeating the EM procedure no more than 1000 times or until the difference between the previous and current estimates of abundances, $\sum_{j=1}^{A} \mid \alpha_j - \alpha'_j \mid$, is less than 0.0001.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

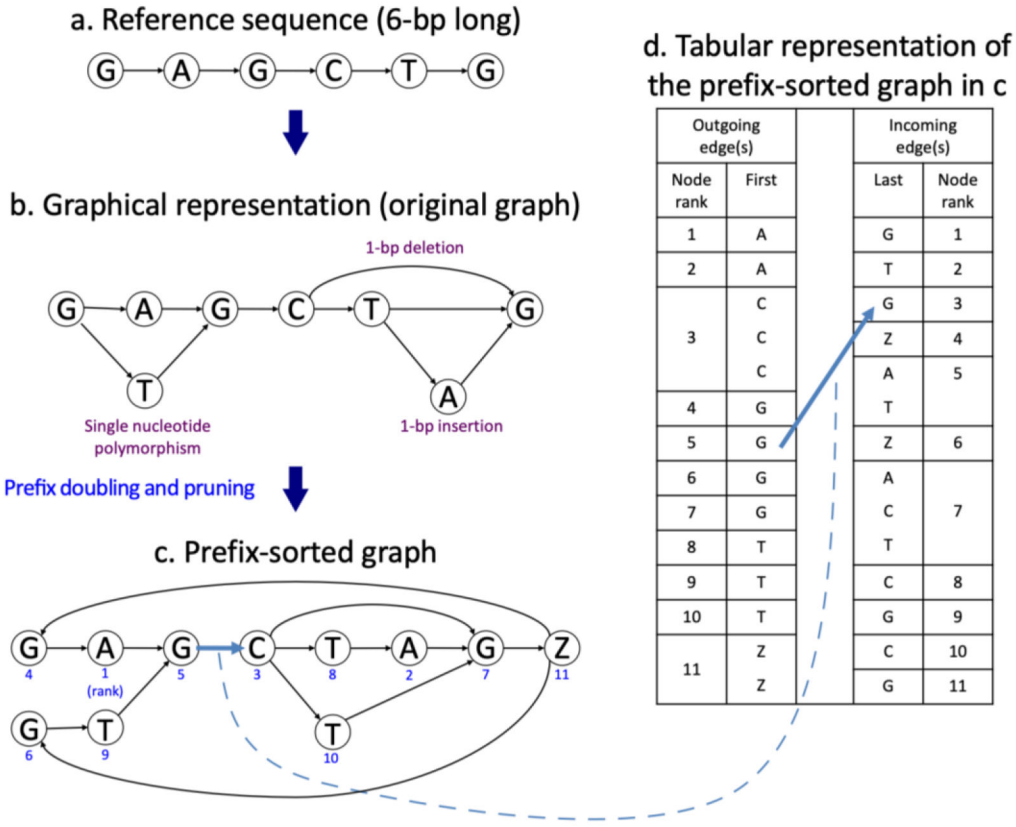
We would like to express our thanks to Kathleen Barnes and Michelle Daya for sharing Omixon's HLA results with us. We would like to thank Ben Langmead and Jacob Pritt for their invaluable contributions to our discussions on HISAT2. We also greatly appreciate the generosity of Gaudenz Danuser and Dana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

References

1. Genomes Project C et al. A map of human genome variation from population-scale sequencing. Nature 467, 1061–1073 (2010). [PubMed: 20981092]

2. Genomes Project C et al. An integrated map of genetic variation from 1,092 human genomes. Nature 491, 56–65 (2012). [PubMed: 23128226]
3. Consortium GT The Genotype-Tissue Expression (GTEx) project. Nat Genet 45, 580–585 (2013). [PubMed: 23715323]
4. Lappalainen T et al. Transcriptome and genome sequencing uncovers functional variation in humans. Nature 501, 506–511 (2013). [PubMed: 24037378]
5. t Hoen PA et al. Reproducibility of high-throughput mRNA and small RNA sequencing across laboratories. Nat Biotechnol 31, 1015–1022 (2013). [PubMed: 24037425]
6. Sanders SJ et al. De novo mutations revealed by whole-exome sequencing are strongly associated with autism. Nature 485, 237–241 (2012). [PubMed: 22495306]
7. Krumm N et al. Excess of rare, inherited truncating mutations in autism. Nat Genet 47, 582–588 (2015). [PubMed: 25961944]
8. Leinonen R, Sugawara H, Shumway M & International Nucleotide Sequence Database, C. The sequence read archive. Nucleic Acids Res 39, D19–21 (2011). [PubMed: 21062823]
9. Lander ES et al. Initial sequencing and analysis of the human genome. Nature 409, 860–921 (2001). [PubMed: 11237011]
10. Sherry ST et al. dbSNP: the NCBI database of genetic variation. Nucleic Acids Research 29, 308–311 (2001). [PubMed: 11125122]
11. Lappalainen I et al. DbVar and DGVa: public archives for genomic structural variation. Nucleic Acids Res 41, D936–941 (2013). [PubMed: 23193291]
12. Burrows MW, D. J. A block sorting lossless data compression algorithm. Digital Equipment Corporation (1994).
13. Ferragina P. M. G. Opportunistic data structures with applications; Proceedings 41st Annual Symposium on Foundations of Computer Science; 2000.
14. Li R, Li Y, Kristiansen K & Wang J SOAP: short oligonucleotide alignment program. Bioinformatics 24, 713–714 (2008). [PubMed: 18227114]
15. Langmead B, Trapnell C, Pop M & Salzberg SL Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10, R25 (2009). [PubMed: 19261174]
16. Li H & Durbin R Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25, 1754–1760 (2009). [PubMed: 19451168]
17. Kent WJ BLAT--the BLAST-like alignment tool. Genome Res 12, 656–664 (2002). [PubMed: 11932250]
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). [PubMed: 18714091]
19. Garrison E et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nat Biotechnol 36, 875–879 (2018). [PubMed: 30125266]
20. Rakocevic G et al. Fast and accurate genomic analyses using genome graphs. Nat Genet 51, 354–362 (2019). [PubMed: 30643257]
21. Green RE et al. A draft sequence of the Neandertal genome. Science 328, 710–722 (2010). [PubMed: 20448178]
22. Lunter G & Goodson M Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. Genome Res 21, 936–939 (2011). [PubMed: 20980556]
23. Degner JF et al. Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. Bioinformatics 25, 3207–3212 (2009). [PubMed: 19808877]
24. Siren J, Valimaki N & Makinen V Indexing Graphs for Path Queries with Applications in Genome Research. Ieee-Acm Transactions on Computational Biology and Bioinformatics 11, 375–388 (2014). [PubMed: 26355784]
25. Kim D, Langmead B & Salzberg SL HISAT: a fast spliced aligner with low memory requirements. Nat Methods 12, 357–360 (2015). [PubMed: 25751142]
26. Robinson J et al. The IPD and IMGT/HLA database: allele variant databases. Nucleic Acids Res 43, D423–431 (2015). [PubMed: 25414341]
27. Hares DR Expanding the CODIS core loci in the United States. Forensic Sci Int Genet 6, e52–54 (2012). [PubMed: 21543275]

28. Luo R et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. Gigascience 1, 18 (2012). [PubMed: 23587118]
29. Compeau PE, Pevzner PA & Tesler G How to apply de Bruijn graphs to genome assembly. Nat Biotechnol 29, 987–991 (2011). [PubMed: 22068540]
30. Li H Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv 1303.3997 (2013).
31. Langmead B & Salzberg SL Fast gapped-read alignment with Bowtie 2. Nat Methods 9, 357–359 (2012). [PubMed: 22388286]
32. Eberle MA 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). [PubMed: 27903644]
33. Erlich RL et al. Next-generation sequencing for HLA typing of class I loci. BMC Genomics 12, 42 (2011). [PubMed: 21244689]
34. Lee H & Kingsford C Kourami: graph-guided assembly for novel human leukocyte antigen allele discovery. Genome Biol 19, 16 (2018). [PubMed: 29415772]
35. Major E, Rigo K, Hague T, Berces A & Juhos S HLA typing from 1000 genomes whole genome and whole exome illumina data. PLoS One 8, e78410 (2013). [PubMed: 24223151]
36. Kessler MD et al. Challenges and disparities in the application of personalized genomic medicine to populations with African ancestry. Nat Commun 7, 12521 (2016). [PubMed: 27725664]
37. Danecek P et al. The variant call format and VCFtools. Bioinformatics 27, 2156–2158 (2011). [PubMed: 21653522]
38. Li H Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics 32, 2103–2110 (2016). [PubMed: 27153593]
39. Pachter L Models for transcript quantification from RNA-Seq. arXiv (2011).
40. Kim D, Song L, Breitwieser FP & Salzberg SL Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res (2016).



Outgoing edge(s)		Incoming edge(s)	
Node rank	First	Last	Node rank
1	A	G	1
2	A	T	2
3	C	G	3
	C	Z	4
	C	A	5
4	G	T	
5	G	Z	6
6	G	A	7
7	G	C	
8	T	T	
9	T	C	8
10	T	G	9
11	Z	C	10
	Z	G	11

Figure 1.

Graph representation of indels and mutations and its tabular representation. Starting with a 6-bp reference sequence, GAGCTG (**a**), the lower graph (**b**) incorporates three variants: a single nucleotide variant (A/T), a 1-bp deletion (T), and a 1-bp insertion (A). A prefix-sorted graph of the graph (**c**) has 11 nodes and 14 edges. Each node has a unique numerical node ID shown in blue to indicate its lexicographical 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 (**d**) has two columns under Outgoing edge(s) that show the node IDs and their labels repeated according to the number of their outgoing edges (i.e. node 3, labeled C, is repeated three times with 3 outgoing edges to nodes 7, 8, and 10, respectively). The table has two columns under Incoming edge(s) that show the node IDs and the 14 labels for the preceding nodes (i.e. 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.

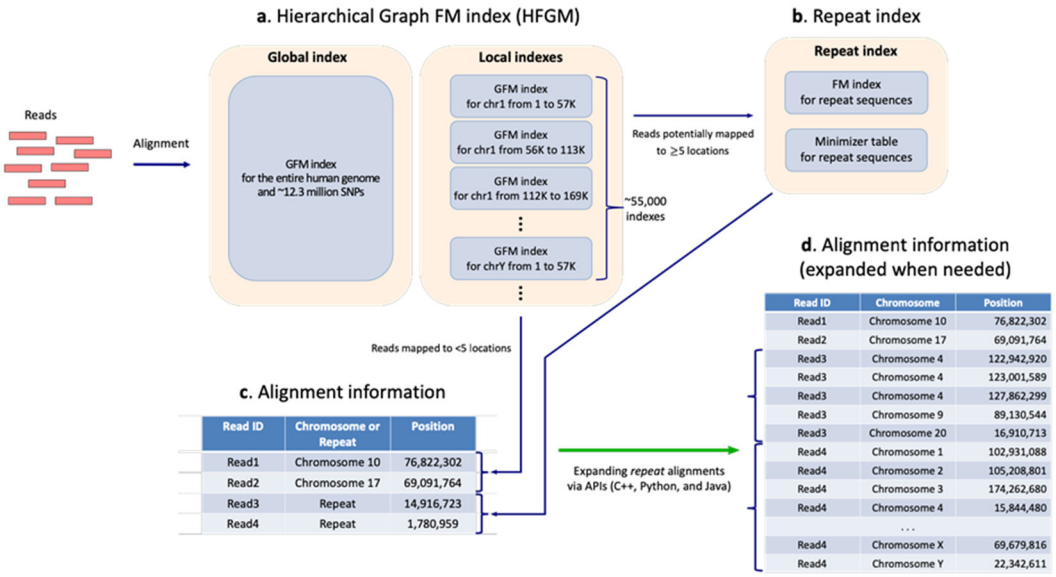


Figure 2.

Overview of HISAT2's indexes and alignment output

(**a**) Hierarchical indexing in the hierarchical graph FM index (HGFM). 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 graph FM indexes, a genome plus a large collection of variants can be searched simultaneously. (**b**) A repeat index represents genomic sequences that are identical. (**c**) A read matching repeat sequences (e.g., *Read3* and *Read4*) is aligned to just one location (the repeat sequence). (**d**) The corresponding genomic locations of repeat aligned reads are retrieved via APIs.

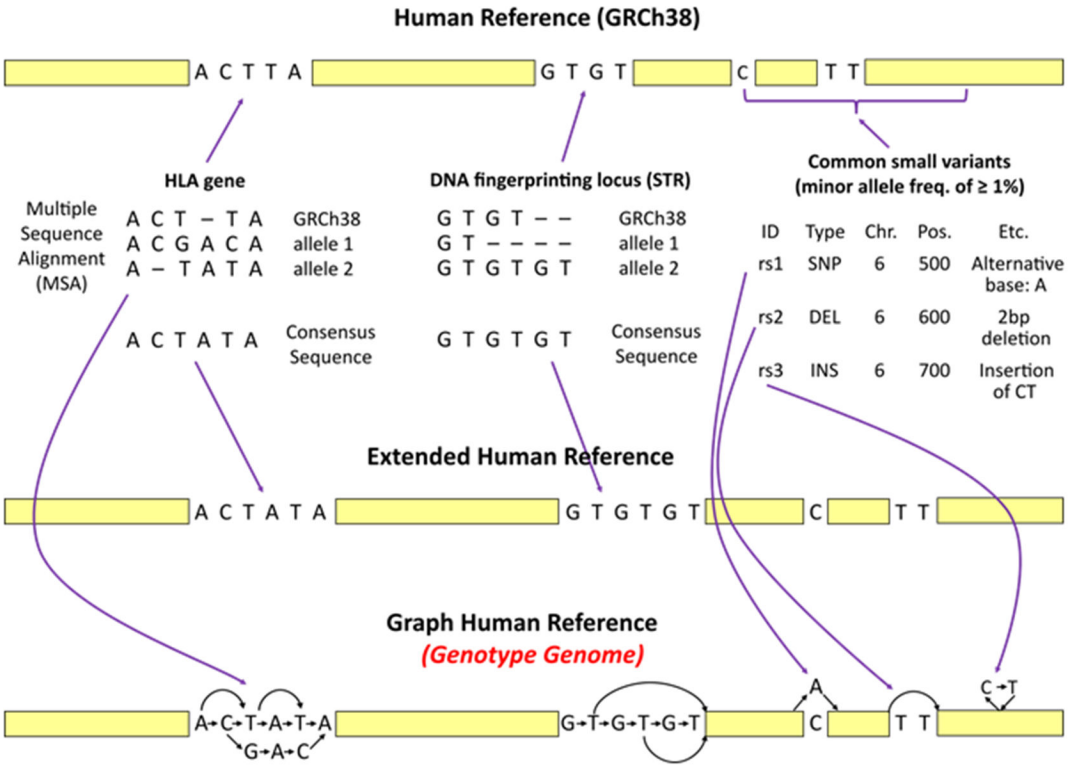


Figure 3.

Construction of the Graph Human Reference, i.e. 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 (i.e. variants with minor allele frequencies of ≥1%) from the dbSNP database. In *a*, 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 based on the most frequent bases that occur in each position of the multiple sequence alignments. The NIST STRBase database 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. In *b*, the human reference is extended by replacing the HLA genes and 13 DNA fingerprinting loci with their consensus sequences. In *c*, the known genomic variants are then incorporated into the extended references using HISAT2's graph data structure. Common small variants from dbSNP such as single nucleotide polymorphisms, deletions, and insertions, are also incorporated into the extended reference. In HISAT-genotype this graph reference is called a *Genotype* genome.

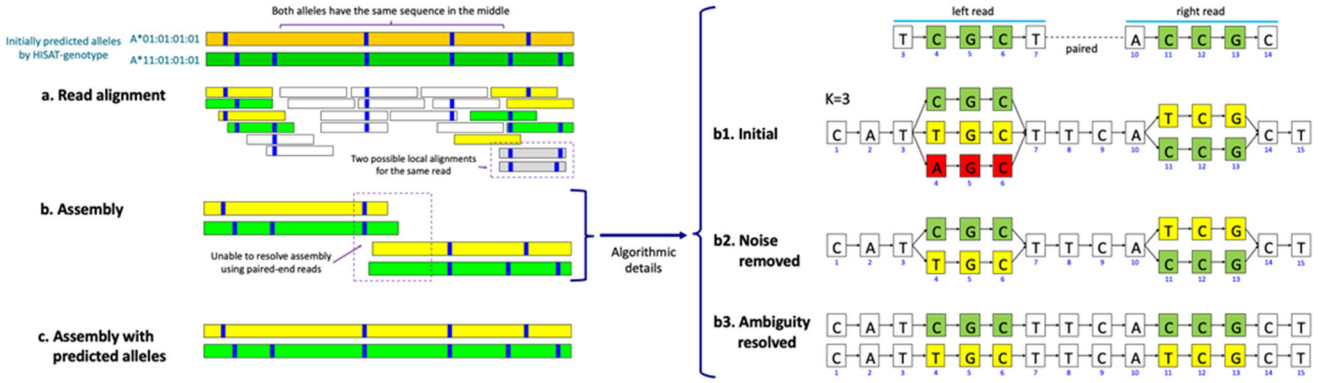


Figure 4.

HISAT-genotype's assembly of two HLA-A alleles through a guided k-mer assembly graph The figure shows an abridged example of HISAT-genotype's assembly output – see Supplementary File 1 for the full assembly output for NA12878. The first two bands are two alleles predicted by HISAT-genotype, in this case A*01:01:01:01 in dark green and 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. (**a**) Shorter bands indicating read alignments whose color is determined according to their 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, i.e. 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. (**b**) Since the two predicted (in fact true/known) alleles share a large common sequence, read pair information is insufficient to fully separate the alleles. HISAT-genotype splits aligned reads into fixed length k-mers. In this simplified case, reads are 5 nucleotides long and k is 3. A pair of reads are aligned at the 3rd location and the 10th location of the graph representation for the HLA gene, respectively. 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. (**c**) In addition, HISAT-genotype uses the predicted alleles to enable full-length assembly of both.

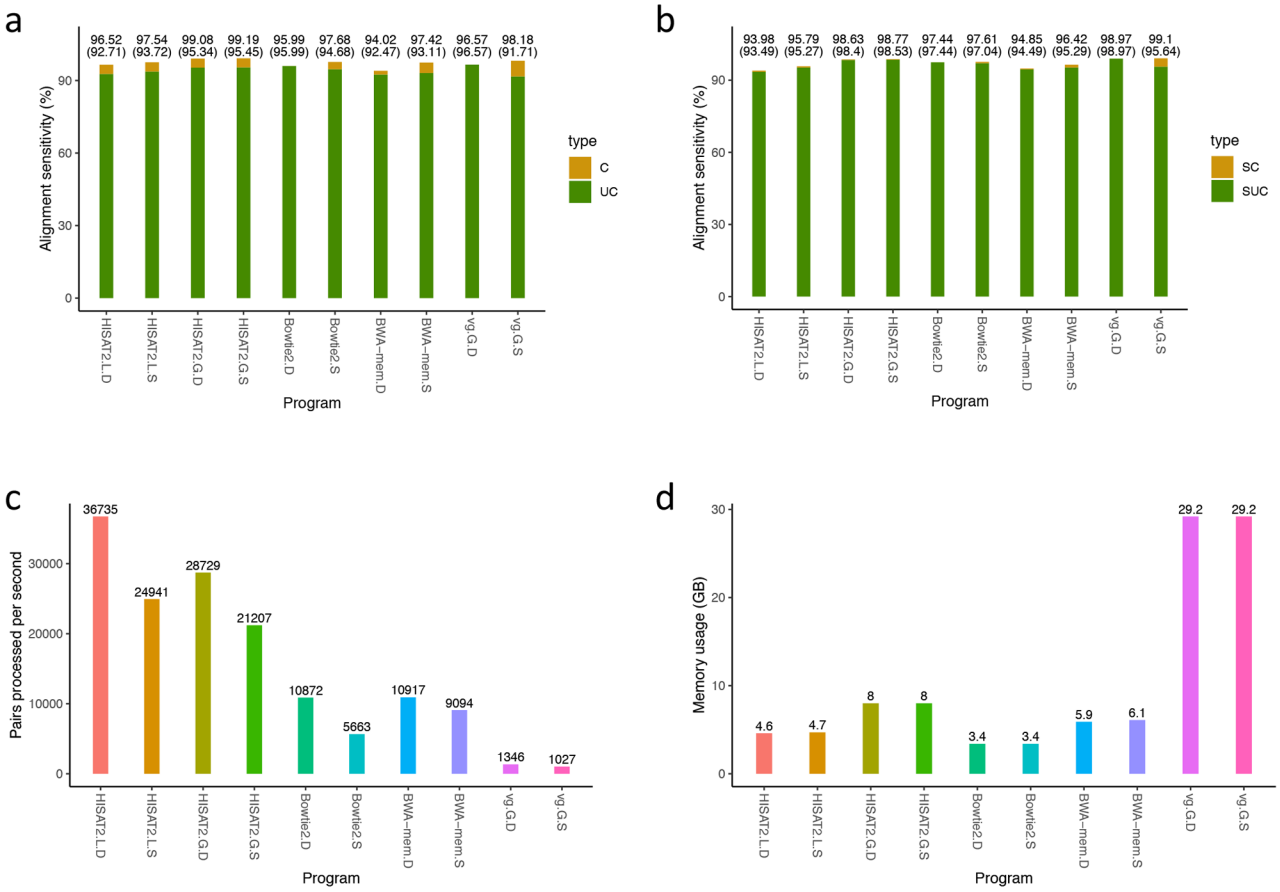


Figure 5.

Comparisons of HISAT2, Bowtie2, BWA-mem, and VG using 10 million simulated read pairs that include SNPs

Alignment sensitivity is defined as the number of correctly aligned read pairs divided by the total number of read pairs.

C: alignment sensitivity calculated based on any one of multiple alignments being correct.

UC: alignment sensitivity calculated based 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.

The suffixes followed by program names stand for as follows: D for default alignment settings, S for sensitive alignment settings, L for linear genome alignment, and G for graph genome alignment.

We ran the programs on the same computer as described in Supplementary Table 7.

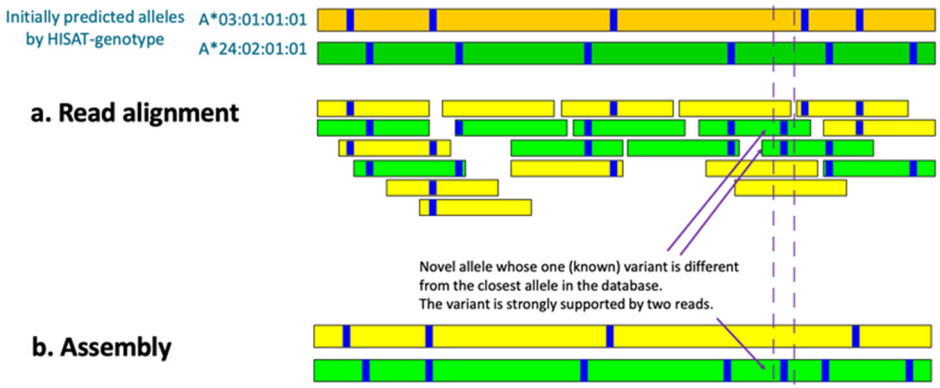


Figure 6.

A 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, A*24:02:01:01. Two reads shown in green support the variant. See Supplementary File 6. for more detailed output from a similar case found in LP6005093-DNA_E03 (a CAAPA genome) at the 2,780th base.

Table 1.

Concordance between HISAT-genotype and Omixon on HLA-typing of 917 genomes from the CAAPA (Consortium on Asthma among African-ancestry Populations in the Americas) collection. Concordance is calculated as the total number of alleles matched between both programs divided by the total number of alleles. For example, for the HLA-A gene, HISAT-genotype and Omixon agree on the allele group (the first number of the HLA type) for both alleles for 913 genomes, agree on one allele for 4 genomes, and agree on no alleles for 0 genomes. Thus, the concordance for HLA-A is 0.998 = (913 × 2 + 4) / (917 × 2). HISAT-genotype reports HLA types with all four fields specified (e.g., A*24:02:01:01), while Omixon reports HLA types with either two numbers (e.g. A*69:01) or three numbers (A*24:02:01); therefore matches were evaluated using only the first two numbers.

	First number (e.g., A*01)				First and second numbers (e.g., A*01:01)			
	Both alleles matched	One allele matched	No allele matched	Concordance	Both alleles matched	One allele matched	No allele matched	Concordance
HLA-A	913	4	0	0.998	883	33	1	0.981
HLA-B	911	6	0	0.997	877	40	0	0.978
HLA-C	915	2	0	0.999	880	34	3	0.978
HLA-DQA1	884	33	0	0.982	868	45	4	0.971
HLA-DQB1	917	0	0	1	753	164	0	0.911
HLA-DRB1	861	56	0	0.97	698	205	14	0.873

Table 2.

HISAT-genotype's DNA fingerprinting results for 17 PG genomes after incorporating 8 novel alleles from the 17 PG data 8 cases where HISAT-genotype and the wet-labs results disagree, yet HISAT-genotype appears to have identified correctly, are indicated in bold and italics and explained in detail in the main text.

		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

	Genome ID																
	NA12877	**NA12878**	**NA12879**	**NA12880**	**NA12881**	**NA12882**	**NA12883**	**NA12884**	**NA12885**	**NA12886**	**NA12887**	**NA12888**	**NA12889**	**NA12890**	**NA12891**	**NA12892**	**NA12893**
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