Published in final edited form as:
Nat Methods. 2024 January ; 21(1): 41–49. doi:10.1038/s41592-023-02069-6.

Improved sequence mapping using a complete reference genome and lift-over

Nae-Chyun Chen[1], **Luis F Paulin**[2], **Fritz J Sedlazeck**[2,3], **Sergey Koren**[4], **Adam M Phillippy**[4], **Ben Langmead**[1]

[1]Department of Computer Science, Johns Hopkins University, Baltimore, MD, 21218, USA

[2]Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX, 77030, USA

[3]Department of Computer Science, Rice University, 6100 Main Street, Houston, TX, USA

[4]Genome Informatics Section, Computational and Statistical Genomics Branch, National Human Genome Research Institute, National Institutes of Health, Bethesda, MD, 20894, USA

Abstract

Complete, telomere-to-telomere genome assemblies promise improved analyses and the discovery of new variants, but many essential genomic resources remain associated with older reference genomes. Thus, there is a need to translate genomic features and read alignments between references. Here we describe a new method called levioSAM2 that accounts for reference changes and performs fast and accurate lift-over between assemblies using a whole-genome map. In addition to enabling the use of multiple references, we demonstrate that aligning reads to a high-quality reference (e.g. T2T-CHM13) and lifting to an older reference (e.g. GRCh38) actually improves the accuracy of the resulting variant calls on the old reference. By leveraging the quality improvements of T2T-CHM13, levioSAM2 reduces small-variant calling errors by 11.4-39.5% compared to GRC-based mapping using real Illumina datasets. LevioSAM2 also improves long-read-based structural variant calling and reduces errors from 3.8-11.8% for a PacBio HiFi dataset. Performance is especially improved for a set of complex medically-relevant genes, where the GRC references are lower quality. The software is available at https://github.com/milkschen/leviosam2 under the MIT license.

1 Introduction

A reference genome serves both as a template for mapping reads and a set of coordinates for interpreting results. Human reference quality has steadily improved in the last decade, with

corresponding author: cnaechy1@jhu.edu; langmea@cs.jhu.edu.

Authors' contributions
NC, SK, AMP, and BL designed the method. NC wrote the software. NC and LFP performed the experiment. NC, LFP, FS, SK, AMP, and BL performed analysis and wrote the manuscript. All authors read and approved the final manuscript.

Availability of data and materials
The software is available at https://github.com/milkschen/leviosam2 under the MIT license. The experiments described in this paper are described at https://github.com/milkschen/levioSAM2-experiments under the MIT license.

Competing interests
The authors declare that they have no competing interests.

over 1,000 GRCh37 issues having been resolved in GRCh38[1], and T2T-CHM13 providing a telomere-to-telomere sequence including difficult-to-assemble regions like segmental duplications[2].

While these improvements can benefit read mapping and downstream analyses[1,3,4], researchers face obstacles when migrating between references. A new reference's coordinate system is incompatible with annotations expressed in the old coordinates. Such annotations might be genotypes[5-7], or functional or phenotype annotations[8-11]. Migrating from GRCh37 to GRCh38 required years of work[12]; even today, many groups report that they have no plans to switch from GRCh37[13].

To facilitate movement between references, tools that "lift" across genomic coordinate systems have been proposed[14-17]. Unfortunately, the lifting process can produce discordant results compared to re-analyzing the sequencing reads from scratch. The problem is particularly pronounced in regions where the references have copy-number differences or assembly artifacts (Figure 1a)[12,18-20]. While the recently described LiftOff method can lift gene annotations with high confidence using re-mapping[21], this strategy works only with genes and not with other types of annotations such as generic intervals, genotypes, or read mappings.

The prior levioSAM software could perform scalable and memory-efficient lift-over of mappings, but did not support complex genomic rearrangements such as translocations and inversions[17]. Other methods either cannot lift read mappings[14,16] or do not scale for large genomic datasets[15].

To make the best use of improved reference assemblies and rich annotations, we propose levioSAM2 for fast and accurate lift-over of read mappings (Figure 1b). In contrast to the more typical strategy of lifting old-reference alignments to a newer reference (old-to-new), we propose to start by mapping to the most complete and error-free assembly like T2T-CHM13, then to use levioSAM2 to lift the mapped reads to an existing annotation-rich reference like GRCh37 or GRCh38. The first step of levioSAM2 is an efficient lift kernel ("levioSAM2-lift") that translates mapping information into the target coordinates. LevioSAM2-lift uses succinct data structures that can update mapped reference name and position information in $O(1)$ time and update the CIGAR alignment string in $O(r + g)$ time, where r is the number of CIGAR runs and g is the number of overlapping chain gaps of a mapping (Figure 1c and Section 4.1).

We further designed a selective strategy, similar to the "reference flow" approach[22], to handle mappings that are influenced by major differences between source and target (Section 4.2). LevioSAM2 classifies lifted reads into three groups. The "suppressed" group consists of reads mapped to regions in the source genome with no counterpart in the target, e.g. the centromeric sequences in T2T-CHM13 and missing collapsed sequences, to avoid false-positive mapping. The "deferred" group consists of reads that can benefit from re-mapping, e.g. because they mapped with low mapping quality. The "committed" group, which generally contains the majority of the reads, consists of reads belonging to neither of the other two groups. Committed reads are mapped with high confidence and are included in the final mapping results (See Section 4.2). LevioSAM2 also collects and reports mappings that are unliftable using the coordinate system of the source reference, enabling analysis in regions unique to the source (Figure S1).

We evaluated levioSAM2 by lifting mappings from T2T-CHM13 to GRCh37 and GRCh coordinates and comparing them to the results obtained by mapping directly to the GRC references. We also called small variants using GATK-HaplotypeCaller[23] and DeepVariant[24]. We observed that levioSAM2 could reduce small variant errors by 11.4% to 39.5% on real Illumina whole-genome-sequencing (WGS) data for HG001, HG002, and HG005 in the Genome in a Bottle (GIAB) v4.2.1 regions[25]. LevioSAM2 yielded larger improvements in the GIAB challenging medically relevant genes (CMRG)[26], reducing small variant errors by 19.4% to 51.3% for HG002. LevioSAM2 also improved mapping of real Pacific Biosciences High Fidelity (PacBio HiFi) reads[27]. Besides achieving improved or comparable small-variant calling accuracy, levioSAM2 reduced structural-variant (SV) errors by 11.8% compared to GRCh38 in the GIAB CMRG regions[26], and reduced SV errors by 3.8% compared to GRCh37 in the GRCh37 GIAB Tier 1 benchmark regions[28].

2 Results

Lifting from T2T-CHM13 improves short-read mapping to GRC references.

We used simulated reads to compare mapping accuracy between the levioSAM2 workflow and a typical single-reference direct mapping method (Section 4.4). We can measure the correctness of a read mapping by comparing its mapping position with its true point of origin according to the simulator. We simulated 10M paired-end 100-bp Illumina reads using mason2[29]. We selected GRCh38 as the “base” reference, where sequences not liftable to CHM13 were masked with the `N` (unknown) symbol during simulation. We also injected the SNPs of HG001 during simulation (see Section 4.4 for details). To assess the influence of the quality of the reference assembly on the mapping process, we performed the simulation using both human chromosome 20 (GRCh38 chr20) and 21 (GRCh38 chr21). Chr21 is known to include 771 kbp of false duplications[26], whereas chr20 has no known false duplications. False duplicates in the reference assembly can “attract” reads from the correct point of origin. We mapped the reads using Bowtie 2[30] and BWA-MEM[31], using default options for both.

The selective levioSAM2 workflow generated an additional 0.08% of correct mappings for chr20 and 1.37% for chr21 versus the direct-to-GRCh38 method (“GRCh38”; Figure S2). The fact that levioSAM2 had a more substantial improvement in chr21 reflects its ability to recover from the false duplicates in GRCh38. Note that since this simulation is based on GRCh38, we are not able to measure whether and how levioSAM2 benefits from assembly improvements in places where GRCh38 has assembly gaps, but characterize these improvements using real data in the following sections.

LevioSAM2 improves short-read small variant calling.

We next evaluated levioSAM2 using two common short-read small variant calling pipelines, BWA-MEM-GATK-HaplotypeCaller[23] and BWA-MEM-DeepVariant[24] (Section 4.5.1). We used real 30× paired-end Illumina Novaseq whole-genome sequencing (WGS) data representing three ancestries (HG001: Utah/European; HG002: Ashkenazi Jewish; HG005: Han Chinese)[32] (Table 1). All three individuals had high-quality genotypes provided by the Genome in a Bottle (GIAB) consortium for both GRCh37 and GRCh38[25]. The GATK-based pipelines that used levioSAM2 to lift reads from T2T-CHM13 to GRCh37 ("CHM13-to-GRCh37") reduced mean small variant error compared to a direct-to-GRCh37 pipeline by 49,770 errors (39.5%). LevioSAM2 also showed an error reduction when lifting from T2T-CHM13 to GRCh38 ("CHM13-to-GRCh38"), avoiding 18,308 (23.9%) small variant errors. The overall F_1 improved as well (Fig 2a and Table S2). We stratified the calls into SNPs and indels and assessed the precision and recall for levioSAM2 and direct-to-GRC pipelines (Fig 2b). LevioSAM2 had higher precision and recall for all samples and both variant types. For all samples, levioSAM2 reduced the mean false-positive error (indel and SNP combined) by 38,981 (52.2%) and 11,551 (24.9%) compared to GRCh37 and GRCh38, respectively (Table S2).

LevioSAM2 also improved small variant calling for the BWA-MEM-DeepVariant approaches. On average, the levioSAM2 workflows avoided 8,778 errors (16.7%) and 3,443 errors (11.4%) compared to GRCh37 and GRCh38, respectively (Fig 2c and Table S3). In contrast to the GATK-based pipelines, the strongest improvement using DeepVariant was for SNP recall, where levioSAM2 reduced mean false-negatives by 8,331 (20.8%) and 4,603 (22.3%) variants compared to GRCh37 and GRCh38 (Fig 2d and Table S3). The discrepancy between the performance of GATK-HaplotypeCaller and DeepVariant could be explained by the difference in their underlying models. The neural network model used by DeepVariant was capable of learning mapping artifacts and reducing false-positive errors in hard-to-map regions (Note S1). The hidden Markov model used by GATK-HaplotypeCaller might not have recognized the mapping artifacts, and thus benefited more from the less-biased mappings generated by levioSAM2.

Both GRCh37 and GRCh38 are known to include false duplications that confound genomic analysis in medically relevant regions. We compared levioSAM2, GRCh37, and GRCh38 for small variant calling performance in the GIAB CMRG regions[26] using the HG002 WGS data. LevioSAM2-based methods were more accurate for both GATK-HaplotypeCaller and DeepVariant pipelines. When using GATK-HaplotypeCaller, the levioSAM2 workflows removed 1,514 (45.4%) and 730 (35.2%) variant calling errors compared to GRCh37 and GRCh38 (Figure 2e and Table S4). When using DeepVariant, levioSAM2 avoided 306 (19.4%) and 1,010 (20.0%) variant calling errors compared to GRCh37 and GRCh38 (Figure 2f and Table S5). The levioSAM2 workflows outperformed direct-to-target methods for both recall and precision for both variant types. LevioSAM2 had a larger improvement in the CMRG regions compared to other regions, again showing that levioSAM2 effectively leverages the improved assembly quality of T2T-CHM13 to improve short-read small variant calling.

We analyzed the reads mapped by CHM13-to-GRCh38 and called small variants in the "unliftable" regions that were unique to T2T-CHM13 using DeepVariant. We called 5,314 variants that passed the default DeepVariant filter, and 1,635 of them were high-quality

$(\texttt{QUAL} \geq 30)$ (Figure S1). These variants were unique to T2T-CHM13 and could not be identified with typical variant calling approaches and GRCh38.

Larger small variant calling improvement in difficult regions.

We investigated differences in small variant calling between levioSAM2 and the GRC references using the GIAB difficult region stratifications[4,26]. These stratify genomic regions by features such as low mappability (“LowMap”), extreme GC content (“GC<25or>65”), low complexity (“Tandem&Homo”), presence of segmental duplications (“SegDups”), and the assembly-artifacts-and-other regions (“OtherDifficult”). The union of these difficult regions comprises the “AllDifficult” regions (see Table S6 for region sizes). We considered only the variant calls within the GIAB confident regions in the evaluation. We observed that levioSAM2 had enriched small variant calling improvements in the LowMap, SegDups, and OtherDifficult regions when using GATK-HaplotypeCaller. In these regions, levioSAM2 had a 47.7%–48.0% and a 27.5%–42.4% reduction in error rate compared to direct-to-GRCh37 and direct-to-GRCh38 respectively. The union of all difficult regions also showed improved small variant calling performance (Figure 3a and Table S7). When using DeepVariant, the most pronounced improvement was in LowMap regions, where levioSAM2 reduced errors by 15.7%–22.3% (Figure S5a and Table S8).

We further analyzed higher-resolution GIAB difficult region strata and ranked them by density of small variant calling error reduction (Figure 3b and Figure S5b). We observed that levioSAM2 reduced errors most in regions with likely reference artifacts (“hs37d5Decoy” and “ChainSelf”, both are subsets of “OtherDifficult”), avoiding up to 971 errors per Mbp compared to direct-to-GRC approaches. Other strata strongly improved by levioSAM2 included regions with large segmental duplications (“SegDups>10kb”), and low-mappability regions comprised of non-unique 250-mers (“NonUnique250bp”) and non-unique 100-mers (“NonUnique100bp”). There were no stratified regions that reported an increased error density of greater than 1/Mbp by levioSAM2 using either GATK-HaplotypeCaller or DeepVariant.

Improved variant calling using PacBio-HiFi long reads.

LevioSAM2 supports lifting alignments spanning chain gaps (Figure 1c), making it suitable for long reads, such as PacBio HiFi reads[27]. We designed a workflow for long reads supporting both minimap2[33] and Winnowmap2[34] (Section 4.2). We mapped a real 28× PacBio-HiFi WGS dataset from HG002[35] to T2T-CHM13 and used levioSAM2 to generate GRC-based mappings. We called small variants using DeepVariant and assessed the calls with the GIAB v4.2.1 truth set. CHM13-to-GRCh37 removed 10,435 small variant errors (9.7%) compared to direct-to-GRCh37. CHM13-to-GRCh38 performed comparably to GRCh38, with 673 (0.8%) more errors (Figure 4a and Table S9). The levioSAM2 workflow generated more accurate small variant calls in the GIAB CMRG regions, where CHM13-to-GRCh37 avoided 30 (1.7%) errors and CHM13-to-GRCh38 removed 337 (22.5%) errors compared to their direct-to-GRC counterparts (Figure 4b and Table S10).

Using the same PacBio-HiFi dataset, we called structural variants (SVs) using Sniffles2[36] and analyzed the results using truvari[37] (Section 4.5.2). We evaluated the SV calls using the GIAB Tier 1 benchmark regions for GRCh37[28] and the GIAB CMRG benchmark for GRCh38[26]. Note that there is not a genome-wide SV benchmark for each of the references. CHM13-to-GRCh37 removed 44 (5.7%) false-positive SV errors (FPs) compared to direct-to-GRCh37 (Figure 4c and Table S11). Most FPs were shared between both methods, and the majority of these were insertions enriched in low-complexity regions. Many of the resolved FPs associated with regions having a many-to-1 mapping from donor to target, or "mapping collapse" (Figure 6b). While CHM13-to-GRCh37 resulted in 4 more false-negative SV errors (FNs) compared to direct-to-GRCh37 using the GIAB Tier 1 SV calls, we observed examples where the GIAB calls did not agree with haplotype-resolved assemblies for the same individual[38] (Figures S6 and S7). We visually inspected these examples and observed evidence of mapping collapse including abnormal coverage and loci with more than two haplotypes, suggesting improved mapping and reduced SV FNs by levioSAM2. Compared to direct-to-GRCh38 in the GIAB CMRG regions, CHM13-to-GRCh38 removed 2 (11.8%) SV calling errors. Both were false deletions in the *KMT2C* gene (Figure 4d and Table S12).

LevioSAM2 reduces large-scale mapping artifacts.

LevioSAM2 can resolve or reduce large-scale mapping artifacts compared to direct-to-GRC pipelines. We first examined the small variant calls using the real 30× HG002 short read dataset. In the medically relevant gene *KMT2C*, we observed high-density mapping errors which had been reported by the GIAB CMRG study[26]. The errors were due to 15-kbp sequences in *KMT2C* in HG002 but collapsed in GRCh37. Therefore, sequences from both regions mapped to *KMT2C*, resulting in an abnormally high mapping depth (up to 296×) and a high alternate allele density within *KMT2C*. The missing homolog was assembled in T2T-CHM13 and marked as a suppressed region by the levioSAM2 annotation workflow. Mapping to T2T-CHM13 correctly placed the reads in the appropriate locus and the levioSAM2 workflow resolved most mapping errors in this region (Figure 5a). We also observed that DeepVariant reported a low number of variants in this region using GRCh37, even when the variant allele frequencies were as high as 0.8. We reasoned that DeepVariant learned this signature of mapping artifacts and suppressed variants in its model (Figure S8 and Note S1).

We then analyzed mappings of the HG002 PacBio-HiFi long read dataset and observed similar large-scale mapping improvements by levioSAM2. Similar to the short-read example in gene *KMT2C*, we observed a high density of alternate alleles in a 56-kbp region which overlapped with another medically relevant gene, *MAP2K3*[39]. We showed that CHM13-to-GRCh37 significantly reduced mapping depth and alternate allele density in this region, suggesting improved mapping. When assessed with the GIAB Tier 1 benchmark for SVs, CHM13-to-GRCh37 avoided a false structural deletion (8.2 kbp) (Figure 5b).

LevioSAM2 is computationally efficient.

LevioSAM2-lift's bitvector-based algorithm is fast and memory-efficient (Figure 6a). Compared to CrossMap[15], levioSAM2-lift used 20.1% of the wall-clock time and 19.9% the peak memory (64.3 MB vs. 341.9 MB) when run on a single thread. Unlike CrossMap, levioSAM2-lift supports multi-thread processing and used only 7.3% wall time and 19.4%

of the memory (66.4 MB vs. 341.9 MB) when using 4 threads (see Figure S4 for thread scaling). LevioSAM2-lift was able to replicate the results of CrossMap, but includes several additional features that are important in practice. For instance, levioSAM2 can lift mappings spanning chain-file gaps and can update CIGAR-string information in the output mappings, enabling lifting of long-read mappings. Further, levioSAM2-lift can optionally update edit distance (`NM:i`) and mismatch encoding alignment string (`MD :z`).

The full levioSAM2 workflow (excluding the initial mapping step to T2T-CHM13) was also faster than a standard approach that simply maps all the 30× Illumina paired-end WGS reads for HG002 to the target reference (Figure 6b). We compared the CPU time used by BWA-MEM[31] when aligning directly to the target and by the samtools sort command[40,41] to the corresponding levioSAM2 run. For a 30× real WGS dataset initially mapped to T2T-CHM13, the levioSAM2 workflows took 36.9 CPU hours for CHM13-to-GRCh37 and 47.6 CPU hours for CHM13-to-GRCh38. Compared to directly mapping to the target genome, the CHM13-to-GRCh37 method took 49.8% of the time (74.1 hours) and the CHM13-to-GRCh38 method took 55.4% of the time (86.3 hours). While other stages in the levioSAM2 workflow used more memory compared to levioSAM2-lift, they didn’t use more than 35 GB memory (Note S2 and Figure S3).

3 Discussion

LevioSAM2 lifts mappings from a source reference to a target reference while selectively remapping the subset of reads for which lifting is not appropriate. LevioSAM2 uses efficient succinct data structures and is much more efficient than existing tools. While it is common to lift post-read-mapping results like variant calls between references[14,15], these translations can be inaccurate due to large-scale differences between references[12,18-20]. Starting from a BAM file mapped to T2T-CHM13, levioSAM2 generated GRC-based mappings 44.7-50.2% faster than a method which remapped all the reads. Mapping and genotyping results after using levioSAM2 were more accurate than those obtained using a single reference in both simulated and real-data scenarios. Since levioSAM2 uses linear references, its results are readily interpretable using common tools like the Integrative Genomics Viewer (IGV)[42].

The small-variant calling improvements by levioSAM2 were enriched in difficult regions with presence of low-mappability units, segmental duplications, and assembly artifacts. These regions include the challenging medically relevant genes (CMRG), where accurate variant calling is known to be difficult. LevioSAM2 also improved long read mapping, demonstrated by more accurate small- and structural-variant calling. Notably, levioSAM2 resolved most mapping errors in a 15-kbp region in the *KMT2C* gene and a 56-kbp region in the *MAP2K3* gene in GRCh37.

As the need to move alignments between assemblies becomes more common, it will be important to improve whole-genome alignment maps between those references. While the UCSC “lift over construction” recipe[14,43] has become a standard approach, more work is needed to assess the quality of the resulting chain files[44]. For example, sequences with multiple copies are challenging to accurately place, and some chain files do not guarantee 1-to-1 mapping between the source and target references. While the levioSAM2 framework tolerates errors in a chain-file alignment, we expect improved whole-genome maps to further enhance the accuracy and computational performance of levioSAM2.

LevioSAM2's decisions on whether to commit, defer or suppress a read mapping rely on relatively simple heuristics. We have shown these are effective in reducing the overall number of needed realignments. Nevertheless, we expect that levioSAM2's accuracy can be further improved by making these decisions more data- and model-driven, perhaps using properties of the input such as the read length, assay type, or data quality.

The future may see a shift toward more sophisticated pangenome representations, e.g. made up of high-quality population-scale genome assemblies[35,45]. We expect that accurate and efficient lift-over methods like levioSAM2 will be useful in these contexts as well. While construction of a pangenome graph can require use of expensive multiple- or progressive-alignment algorithms, the levioSAM2 approach can be applied to any pair of genomes with a whole-genome alignment to each other (in the form of a chain file).

Rapid progress in genome assembly and sequencing is making hundreds of high-quality reference assemblies available[35]. These assemblies provide demonstrated improvements for short and long-read variant calling[2,4]. However, a rich set of annotations, key for interpretation, are built on previous references and difficult or impossible to translate to every new assembly. LevioSAM2 enables analyses that have the best of both worlds by improving mapping for the original reference without losing all its secondary information while also providing mappings for novel genomic discovery in regions unique to the new reference.

4 Methods

4.1 Efficient lift-over using succinct data structures

A chain file describes a pairwise whole-genome alignment[14]. It consists of many "chains", each a set of co-linear alignments between the source and the target reference. An alignment is further split into an interleaved sequence of aligned segments and gaps. An aligned segment can include matches and mismatches but not gaps. A gap can appear in one of the references or both. Each line in a chain specifies one aligned segment and up to two gaps (Figure 1c: bottom).

LevioSAM2 first sorts the aligned segments by position and stores them in a chain interval array. Each chain interval records information useful for lifting, including target contig, strand, and offset. The offset is represented as the difference between source position and target position.

To enable queries against the chain interval array, levioSAM2 builds a pair genome-length of succinct bit vectors ("BV") (Fig. 1c: top). It uses the `start_bv` bit vector to encode starting positions of all chain intervals and uses the `end_bv` vector for ending positions. Both bit vectors are supplemented with data structures enabling constant-time rank queries, as provided by the SDSL library[46]. The levioSAM2 implementation further wraps these bit vectors and arrays in an unordered map, with source contigs as keys to the map.

When querying a position, levioSAM2 first locates the index of the corresponding chain interval by performing a rank query over both `start_bv` and `end_bv`. A rank query computes the number of set bits (bits equal to 1) prior to the queried position. If a position p is within a chain interval, it must be that $\mathtt{rank}_{\mathtt{start\_bv}}(p) - \mathtt{rank}_{\mathtt{end\_bv}}(p) = 1$. For a position outside of any chain interval, levioSAM2 checks the distance between the position and its neighbor chain interval boundary. If the distance is under a user-defined threshold, the query is assigned to the neighbor. LevioSAM2 queries the chain interval array using the index and updates the contig, strand and position information. Since it is dominated by the rank query, this is a constant-time algorithm $(O(1))$, which is faster than the commonly-used interval tree-based algorithms[14,15], which can use $O(\log(m))$ time m is the number of intervals.

It is necessary to update the CIGAR information of an alignment when it overlaps a chain gap. This kind of CIGAR update is not performed by CrossMap[15]. While levioSAM does perform such an update[17], this takes $O(n)$ time where n is the length of a read. Here we describe a new algorithm (Alg. 1) to update the CIGAR string. The algorithm maintains a list of chain gaps, and updates the chain gaps into the CIGAR string. Since the algorithm only traverses through the CIGAR string and gaps collection once, its time complexity is $O(r + g)$, where r is the number of runs in the original CIGAR and g is the number of overlapped chain gaps in the alignment. Usually r and g are much smaller than the read length n, so the proposed algorithm is substantially faster than levioSAM's.

While levioSAM2 can lift over mappings spanning gaps in a chain file and update the CIGAR strings accordingly, the resulting mapping does not always align optimally after lift-over. That is, a slightly different decision about how to arrange gaps and mismatches might be more optimal. To achieve optimal alignment, levioSAM2 includes an optional realignment module that uses a localized dynamic-programming algorithm to refine the alignment. Note that we use the term "mapping" for the task of determining where the read camp from, and the term "alignment" for the more detailed task of lining individual read bases up with reference bases. We use the ksw2 library[47] for efficient dynamic-programming-based realignments. We support parameter presets in the YAML format[48] for popular aligners including Bowtie 2, BWA MEM, and minimap2. LevioSAM2 uses the data structures provided by htslib[49] to process SAM and BAM files.

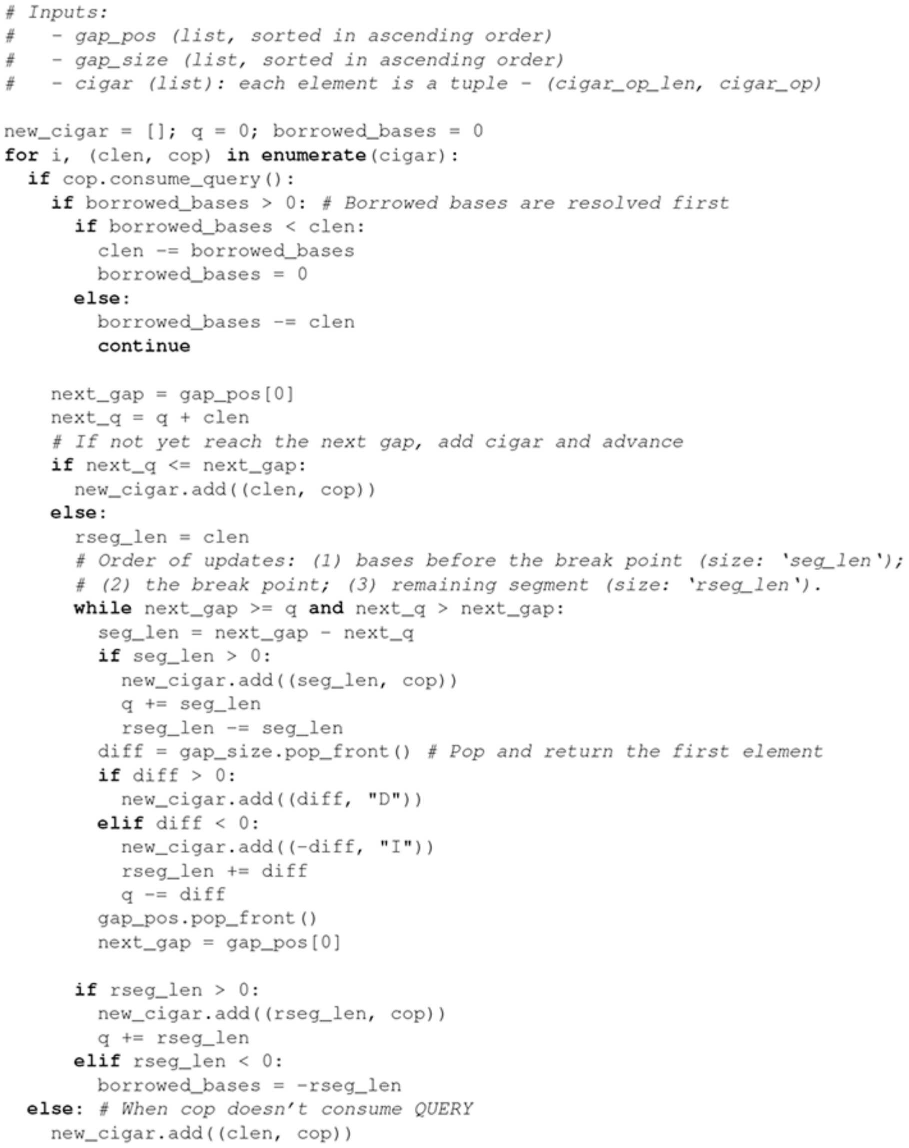
4.2 LevioSAM2 workflow with selective re-mapping

To improve mappings in the presence of large-scale differences between source and target references, levioSAM2 uses a selective re-mapping strategy. Reads that can be lifted with high confidence are "committed" and the lifted alignment is taken as the final alignment. Reads belonging to a region that is specific to the source reference are "suppressed" and left unaligned with respect to the target reference. Reads for which the lift is lower-confidence are "deferred" and are re-mapped to the target reference (Fig. 1b; Section 4.2.1). For paired-end reads, we use the "levioSAM2-collate" step to ensure deferred reads are re-mapped as a pair (Section 4.2.2). After re-mapping, "levioSAM2-reconcile" compares the re-mapped and lifted mappings for each deferred read and selects the one with higher confidence (Section 4.2.3). Committed and reconciled mappings are combined in the final output.

4.2.1 Selective strategy—

To determine if a given read should be committed, levioSAM2 examines a combination of the read's alignment features and the reference genome's "liftability" annotation. Alignment features include mapping status (mapped or not), mapping quality, fraction of clipped bases, edit distance (the `NM:i tag`), alignment score (the `AS :i tag`), and fragment length (if part of a pair). For BWA-MEM (local alignment), we used a MAPQ cutoff of 30 or an alignment score cutoff of 100; for Bowtie 2 (end-to-end alignment), we used a MAPQ cutoff of 10 or an alignment score cutoff of −10.

Algorithm 1: UpdateCigar



```
# Inputs:
#   - gap_pos (list, sorted in ascending order)
#   - gap_size (list, sorted in ascending order)
#   - cigar (list): each element is a tuple - (cigar_op_len, cigar_op)

new_cigar = []; q = 0; borrowed_bases = 0
for i, (clen, cop) in enumerate(cigar):
  if cop.consume_query():
    if borrowed_bases > 0: # Borrowed bases are resolved first
      if borrowed_bases < clen:
        clen -= borrowed_bases
        borrowed_bases = 0
      else:
        borrowed_bases -= clen
        continue

    next_gap = gap_pos[0]
    next_q = q + clen
    # If not yet reach the next gap, add cigar and advance
    if next_q <= next_gap:
      new_cigar.add((clen, cop))
    else:
      rseg_len = clen
      # Order of updates: (1) bases before the break point (size: 'seg_len');
      # (2) the break point; (3) remaining segment (size: 'rseg_len').
      while next_gap >= q and next_q > next_gap:
        seg_len = next_gap - next_q
        if seg_len > 0:
          new_cigar.add((seg_len, cop))
          q += seg_len
          rseg_len -= seg_len
        diff = gap_size.pop_front() # Pop and return the first element
        if diff > 0:
          new_cigar.add((diff, "D"))
        elif diff < 0:
          new_cigar.add((-diff, "I"))
          rseg_len += diff
          q -= diff
        gap_pos.pop_front()
        next_gap = gap_pos[0]

      if rseg_len > 0:
        new_cigar.add((rseg_len, cop))
        q += rseg_len
      elif rseg_len < 0:
        borrowed_bases = -rseg_len
  else: # When cop doesn't consume QUERY
    new_cigar.add((clen, cop))
```

The genomic liftability annotation is a BED file marking some regions as "unliftable" and some as "mappability-reduced." The annotation is generated ahead of the lift-over process and is indexed using interval trees for fast queries. The "unliftable" annotation is given to regions that are unique to the source reference, like T2T-CHM13's centromeric regions, which have no counterpart in GRCh38. Reads mapping to these regions are suppressed in order to avoid false re-mappings. Suppression tends to improve computational performance as well, since suppressed reads often come from repetitive regions and require a disproportionate amount of alignment effort. To determine which regions are unliftable, we first extract sequences from the source reference that are not in the chain file and are longer than 5000 bps. We then align the sequences to the target reference using Winnowmap2[34] and label either unmapped or repetitive (mapping to multiple loci in target) regions as unliftable.

LevioSAM2 also looks for "mappability-reduced" regions in the source to gauge confidence in lifted mappings. Mappability-reduced regions have high mappability in the source reference but lower mappability after being mapped to the target. To build the mappability-reduced annotations, we use GenMap2[50] to calculate mappability for both source and target references (using 100-mers and a 0.01 mismatch-rate tolerance). We extract uniquely mappable regions in the source reference and lift them to the target. We then overlap these lifted unique regions with the low-mappability regions in the target. Reads lifted to these regions are deferred and re-mapped.

4.2.2 Collate—

For paired-end reads, the selective strategy can result in cases where the paired ends are assigned to different groups. Re-mapping accuracy for single-end reads is usually lower than for paired-end reads, especially when the reads overlap indels. Thus, we develop the "levioSAM2-collate" method to additionally defer the mate of the deferred singletons, generating properly paired deferred read sets.

LevioSAM2-collate starts by reading the deferred alignments and storing the first and the second segments separately in a pair of hash maps[51]. For a properly-paired read (i.e. with both ends in the deferred group), we write both ends to a paired-deferred BAM file and remove them from the hash tables. Once all deferred alignments have been processed, any paired ends remaining in the hash maps are singletons. We then read the committed alignments and extract the reads which pair with the deferred singletons.

4.2.3 Reconciling—

Motivated by the "reference flow" mapping strategy[22], we designed a "levioSAM2-reconcile" method to improve accuracy. LevioSAM2-reconcile compares the lifted and re-mapped deferred mappings, selecting the one which has higher confidence. The process starts with sorting both deferred BAM files by query name. We select the mapping with lower edit distance with respect to the target reference. Ties are broken by taking the mapping with higher mapping quality, or by making a random choice if they are still tied. By reconciling and selecting the alignment in this way, levioSAM2 can better leverage the genetic diversity provided by the additional reference.

4.3 Generating chain files

We used nf-LO[43] with the minimap2 mode to build the chain files for both CHM13-to-GRCh37 and CHM13-to-GRCh38:

```
nextflow run main.nf --source source.fa --target target.fa \
    --outdir out_dir -profile local --aligner minimap2
```

4.4 Evaluation using simulated sequencing datasets

We used the GRCh38-based SNPs for NA12878/HG001 from the 1000 Genomes Project (1KGP)[12] to build personalized chromosome 20 and 21 references for simulation. We masked the regions that could not lift to T2T-CHM13 using bedtools-maskfasta[52]. We used the mason simulator[29] to simulate 10M 100-bp paired-end reads (5M pairs) for both chromosomes:

```
mason_simulator --num-threads 16 -ir ref.fa -n 5000000 \
    -o out-R1.fq -or out-R2.fq -oa out.sam -iv sample.vcf
```

We mapped the reads using both unmasked GRCh38 and CHM13-to-GRCh38 using default parameters for both Bowtie 2[30] and BWA-MEM[31]. We compared mappings from the direct-to-GRCh38 and CHM13-to-GRCh38 strategies by measuring the fraction of correct mappings, where a mapping was considered correct if its leftmost mapped position (the first un-clipped base) was within 10-bp of that of its simulated origin.

4.5 Evaluation using real sequencing datasets

4.5.1 Small variant calling and evaluation—We evaluated three real 30× coverage datasets sequenced using Illumina Novaseq and a PCR-free protocol. We also evaluated a dataset sequenced with PacBio-HiFi with 28× coverage (Table 1). The samples were from distinct ancestries, including Utah/European (HG001), Ashkenazi Jewish (HG002), and Han Chinese (HG005).

For the Illumina datasets, we mapped the reads to GRCh37, GRCh38, and T2T-CHM13 using BWA-MEM[31] with default parameters and sorted by genomic positions using samtools[41]. We used the levioSAM2 workflow to lift the reads mapped to T2T-CHM13 to GRC references and generated CHM13-to-GRCh37 and CHM13-to-GRCh38 mappings. We then called small variants using GATK HaplotypeCaller[23] and DeepVariant (the "WGS" model)[24] for GRCh37, GRCh38, CHM13-to-GRCh37, and CHM13-to-GRCh38. For the PacBio-HiFi dataset, we followed similar procedures, except we used minimap2[33] as the read mapper and called small variants using the haplotype-sorting DeepVariant pipeline with the "PACBIO" model[53,54].

For both sequencing data types, we evaluated the accuracy of small variants using hap.py[55] and the Genome in a Bottle (GIAB) v4.2.1[25] and the GIAB Challenging Medically Relevant Gene (CMRG) benchmark[26] truth sets. Hap.py reports small variant calling accuracy measures stratified by variant type. We sometimes reported the overall (SNP and indel) F_1 of one dataset for simplicity. We did so by adding the true positives (TP), false positives (FP), and false negatives (FN) for SNPs and indels, then calculating overall F_1

with $2 \cdot \mathrm{TP} / (\mathrm{TP} + 0.5(\mathrm{FP} + \mathrm{FN}))$. Similarly, for the F_1 of multiple datasets, we summed the measures for all datasets and calculated the F_1.

4.5.2 Structural variant calling and evaluation—

We called structural variants (SV) for the PacBio-HiFi HG002 dataset using Sniffles2 v2.0.1[36]. We used the "germline SV calling" mode with default parameters, without providing any tandem repeat annotations. We next compressed the VCF files for each dataset using bgzip and indexed them with tabix[49]. Finally, we benchmarked and compared the SV calls using the GIAB Tier 1 benchmark regions for GRCh37[28] and the GIAB CMRG benchmark for GRCh38[26] using truvari 2.1[37] and following the GIAB benchmarking instructions.

```
sniffles2.0.1.py \
    --input sample.bam \
    --vcf sample.vcf \
    --threads 8 \
    --output-rnames \
    --sample-id sample_id
truvari-v2.1 bench \
    --base grch37-HG002_SVs_Tier1_v0.6.vcf.gz \
    --comp sample.vcf.gz \
    --output sample_id-GIABv0.6 \
    --passonly \
    --includebed grch37-HG002_SVs_Tier1_v0.6.bed \
    --refdist 2000 \
    --reference grch37.fasta \
    --giabreport
truvari-v2.1 bench \
    --base HG002_GRCh38_difficult_medical_gene_SV_benchmark_v0.01.vcf.gz \
    --comp sample.vcf.gz \
    --output sample_id-vs-GRCh38_SV_v0.01.03 \
    --multimatch \
    --passonly \
    --refdist 2000 \
    --includebed HG002_GRCh38_difficult_medical_gene_SV_benchmark_v0.01.bed \
    --reference grch38.fa
```

4.6 Computational efficiency measurement

The computing nodes we used were Intel Cascade Lake 6248R with 48 cores per node and 192 GB of memory. We requested 36 cores for all experiments except for the thread scaling experiments. We used GNU Time[56] to measure CPU time ("User time" + "System time"), wall clock time ("Elapsed (wall clock) time") and peak memory usage ("Maximum resident set size").

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

Part of this research project was conducted using computational resources at the Maryland Advanced Research Computing Center (MARCC). Pre-built levioSAM2 resources for T2T-CHM13 to GRC references are made freely available on Amazon Web Services thanks to the AWS Public Dataset Program. We thank Taher Mun for his advice and contribution to the levioSAM2 programming infrastructure. We appreciate advice from Hann-Chyun Chen on software deployment, Christopher Pockrandt on mappability resources, Alaina Shumate on gene lift-over, and Samantha Zarate on T2T-CHM13 variant analysis. We also thank Andrew Carroll and Pi-Chuan Chang for DeepVariant discussions, Arang Rhie for T2T-CHM13 discussions, and Justin Zook for GIAB strata suggestions.

Funding

NC and BL were supported by NIH grants R01HG011392 and R35GM139602 to BL. FJS and LP were supported by NIH grants 1U01HG011758-01 and UM1HG008898. SK and AMP were supported by the Intramural Research Program of the National Human Genome Research Institute (NHGRI), National Institutes of Health (NIH).

References

1. Schneider VA, Graves-Lindsay T, Howe K, Bouk N, Chen H-C, Kitts PA, Murphy TD, Pruitt KD, Thibaud-Nissen F, Albracht D, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. Genome research 27, 849–864 (2017). [PubMed: 28396521]
2. Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze AV, Mikheenko A, Vollger MR, Altemose N, Uralsky L, Gershman A, et al. The complete sequence of a human genome. Science 376, 44–53 (2022). [PubMed: 35357919]
3. Guo Y, Dai Y, Yu H, Zhao S, Samuels DC & Shyr Y Improvements and impacts of GRCh38 human reference on high throughput sequencing data analysis. Genomics 109, 83–90 (2017). [PubMed: 28131802]
4. Aganezov S, Yan SM, Soto DC, Kirsche M, Zarate S, Avdeyev P, Taylor DJ, Shafin K, Shumate A, Xiao C, et al. A complete reference genome improves analysis of human genetic variation. bioRxiv (2021).
5. 1000 Genomes Project Consortium et al. A global reference for human genetic variation. Nature 526, 68 (2015). [PubMed: 26432245]
6. Karczewski KJ, Francioli LC, Tiao G, Cummings BB, Alföldi J, Wang Q, Collins RL, Laricchia KM, Ganna A, Birnbaum DP, et al. The mutational constraint spectrum quantified from variation in 141,456 humans. Nature 581, 434–443 (2020). [PubMed: 32461654]
7. Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, Downey P, Elliott P, Green J, Landray M, et al. UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS medicine 12, e1001779 (2015). [PubMed: 25826379]
8. Smigielski EM, Sirotkin K, Ward M & Sherry ST dbSNP: a database of single nucleotide polymorphisms. Nucleic acids research 28, 352–355 (2000). [PubMed: 10592272]
9. Mailman MD, Feolo M, Jin Y, Kimura M, Tryka K, Bagoutdinov R, Hao L, Kiang A, Paschall J, Phan L, et al. The NCBI dbGaP database of genotypes and phenotypes. Nature genetics 39, 1181–1186 (2007). [PubMed: 17898773]
10. Taliun D, Harris DN, Kessler MD, Carlson J, Szpiech ZA, Torres R, Taliun SAG, Corvelo A, Gogarten SM, Kang HM, et al. Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. Nature 590, 290–299 (2021). [PubMed: 33568819]
11. Consortium G The GTEx Consortium atlas of genetic regulatory effects across human tissues. Science 369, 1318–1330 (2020). [PubMed: 32913098]
12. Lowy-Gallego E, Fairley S, Zheng-Bradley X, Ruffier M, Clarke L, Flicek P, 1000 Genomes Project Consortium, et al. Variant calling on the GRCh38 assembly with the data from phase three of the 1000 Genomes Project. Wellcome Open Research 4 (2019).

13. Lansdon LA, Cadieux-Dion M, Yoo B, Miller N, Cohen AS, Zellmer L, Zhang L, Farrow EG, Thiffault I, Repnikova EA, et al. Factors Affecting Migration to GRCh38 in Laboratories Performing Clinical Next-Generation Sequencing. The Journal of Molecular Diagnostics 23, 651–657 (2021). [PubMed: 33631350]

14. Fujita PA, Rhead B, Zweig AS, Hinrichs AS, Karolchik D, Cline MS, Goldman M, Barber GP, Clawson H, Coelho A, et al. The UCSC genome browser database: update 2011. Nucleic acids research 39, D876–D882 (2010). [PubMed: 20959295]

15. Zhao H, Sun Z, Wang J, Huang H, Kocher J-P & Wang L CrossMap: a versatile tool for coordinate conversion between genome assemblies. Bioinformatics 30, 1006–1007 (2014). [PubMed: 24351709]

16. Picard toolkit https://broadinstitute.github.io/picard/. 2019.

17. Mun T, Chen N-C & Langmead B LevioSAM: Fast lift-over of variant-aware reference alignments. Bioinformatics (2021).

18. Pan B, Kusko R, Xiao W, Zheng Y, Liu Z, Xiao C, Sakkiah S, Guo W, Gong P, Zhang C, et al. Similarities and differences between variants called with human reference genome HG19 or HG38. BMC bioinformatics 20, 17–29 (2019). [PubMed: 30626316]

19. Ormond C, Ryan NM, Corvin A & Heron EA Converting single nucleotide variants between genome builds: from cautionary tale to solution. Briefings in Bioinformatics (2021).

20. Li H, Dawood M, Khayat MM, Farek JR, Jhangiani SN, Khan ZM, Mitani T, Coban-Akdemir Z, Lupski JR, Venner E, et al. Exome variant discrepancies due to reference genome differences. The American Journal of Human Genetics (2021).

21. Shumate A & Salzberg SL Liftoff: accurate mapping of gene annotations. Bioinformatics 37, 1639–1643 (2021). [PubMed: 33320174]

22. Chen N-C, Solomon B, Mun T, Iyer S & Langmead B Reference flow: reducing reference bias using multiple population genomes. Genome biology 22, 1–17 (2021). [PubMed: 33397451]

23. Poplin R, Ruano-Rubio V, DePristo MA, Fennell TJ, Carneiro MO, Van der Auwera GA, Kling DE, Gauthier LD, Levy-Moonshine A, Roazen D, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. BioRxiv, 201178 (2018).

24. Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, et al. A universal SNP and small-indel variant caller using deep neural networks. Nature biotechnology 36, 983 (2018).

25. Wagner J, Olson ND, Harris L, Khan Z, Farek J, Mahmoud M, Stankovic A, Kovacevic V, Wenger AM, Rowell WJ, et al. Benchmarking challenging small variants with linked and long reads. BioRxiv (2020).

26. Wagner J, Olson ND, Harris L, McDaniel J, Cheng H, Fungtammasan A, Hwang Y-C, Gupta R, Wenger AM, Rowell WJ, et al. Curated variation benchmarks for challenging medically relevant autosomal genes. Nature Biotechnology, 1–9 (2022).

27. Wenger AM, Peluso P, Rowell WJ, Chang P-C, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nature biotechnology 37, 1155–1162 (2019).

28. Zook JM, Hansen NF, Olson ND, Chapman L, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, et al. A robust benchmark for detection of germline large deletions and insertions. Nature biotechnology 38, 1347–1355 (2020).

29. Holtgrewe M. Mason: a read simulator for second generation sequencing data. Technical Reports of Institut für Mathematik und Informatik, Freie Universität Berlin **TR-B-10-06** (2010).

30. Langmead B & Salzberg SL Fast gapped-read alignment with Bowtie 2. Nature methods 9, 357 (2012). [PubMed: 22388286]

31. Li H Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:1303.3997 (2013).

32. Baid G, Nattestad M, Kolesnikov A, Goel S, Yang H, Chang P-C & Carroll A An extensive sequence dataset of gold-standard samples for benchmarking and development. bioRxiv (2020).

33. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34, 3094–3100 (2018). [PubMed: 29750242]

34. Jain C, Rhie A, Hansen NF, Koren S & Phillippy AM Long-read mapping to repetitive reference sequences using Winnowmap2. Nature Methods, 1–6 (2022). [PubMed: 35017739]
35. Jarvis ED, Formenti G, Rhie A, Guarracino A, Yang C, Wood J, Tracey A, Thibaud-Nissen F, Vollger MR, Porubsky D, et al. Automated assembly of high-quality diploid human reference genomes. bioRxiv (2022).
36. Smolka M, Paulin LF, Grochowski CM, Mahmoud M, Behera S, Gandhi M, Hong K, Pehlivan D, Scholz SW, Carvalho CM, et al. Comprehensive Structural Variant Detection: From Mosaic to Population-Level. bioRxiv (2022).
37. English AC, Menon VK, Gibbs R, Metcalf GA & Sedlazeck FJ Truvari: Refined Structural Variant Comparison Preserves Allelic Diversity. bioRxiv (2022).
38. Cheng H, Concepcion GT, Feng X, Zhang H & Li H Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. Nature Methods 18, 170–175 (2021). [PubMed: 33526886]
39. Mandelker D, Schmidt RJ, Ankala A, McDonald Gibson K, Bowser M, Sharma H, Duffy E, Hegde M, Santani A, Lebo M, et al. Navigating highly homologous genes in a molecular diagnostic setting: a resource for clinical next-generation sequencing. Genetics in Medicine 18, 1282–1289 (2016). [PubMed: 27228465]
40. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G & Durbin R The sequence alignment/map format and SAMtools. Bioinformatics 25, 2078–2079 (2009). [PubMed: 19505943]
41. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, et al. Twelve years of SAMtools and BCFtools. Gigascience 10, giab008 (2021). [PubMed: 33590861]
42. Thorvaldsdóttir H, Robinson JT & Mesirov JP Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in bioinformatics 14, 178–192 (2013). [PubMed: 22517427]
43. Talenti A & Prendergast J nf-LO: A Scalable, Containerized Workflow for Genome-to-Genome Lift Over. Genome Biology and Evolution 13, evab183 (2021). [PubMed: 34383887]
44. Garrison E & Guarracino A Unbiased pangenome graphs. bioRxiv (2022).
45. Ebert P, Audano PA, Zhu Q, Rodriguez-Martin B, Porubsky D, Bonder MJ, Sulovari A, Ebler J, Zhou W, Mari RS, et al. Haplotype-resolved diverse human genomes and integrated analysis of structural variation. Science (2021).
46. Gog S, Beller T, Moffat A & Petri M From Theory to Practice: Plug and Play with Succinct Data Structures in 13th International Symposium on Experimental Algorithms, (SEA 2014) (2014), 326–337.
47. Li H Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics 32, 2103–2110 (2016). [PubMed: 27153593]
48. Rapid YAML https://github.com/biojppm/rapidyaml. 2022.
49. Bonfield JK, Marshall J, Danecek P, Li H, Ohan V, Whitwham A, Keane T & Davies RM HTSlib: C library for reading/writing high-throughput sequencing data. GigaScience 10, giab007 (2021). [PubMed: 33594436]
50. Pockrandt C, Alzamel M, Iliopoulos CS & Reinert K GenMap: ultra-fast computation of genome mappability. Bioinformatics 36, 3687–3692 (2020). [PubMed: 32246826]
51. Leitner-Ankerl M. Robin Hood Unordered Map and Set https://github.com/martinus/robin-hood-hashing. 2022.
52. Quinlan AR & Hall IM BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26, 841–842 (2010). [PubMed: 20110278]
53. Martin M, Patterson M, Garg S, Fischer SO, Pisanti N, Klau GW, Schöenhuth A & Marschall T WhatsHap: fast and accurate read-based phasing. BioRxiv, 085050 (2016).
54. Cook D, Kolesnikov A, Chang P-C & Carroll A Improving Variant Calling using Haplotype Information https://google.github.io/deepvariant/posts/2021-02-08-the-haplotype-channel/. 2021.
55. Krusche P, Trigg L, Boutros PC, Mason CE, Francisco M, Moore BL, Gonzalez-Porta M, Eberle MA, Tezak Z, Lababidi S, et al. Best practices for benchmarking germline small-variant calls in human genomes. Nature biotechnology 37, 555–560 (2019).

56. Gordon A. GNU Time https://www.gnu.org/software/time/. 2018.

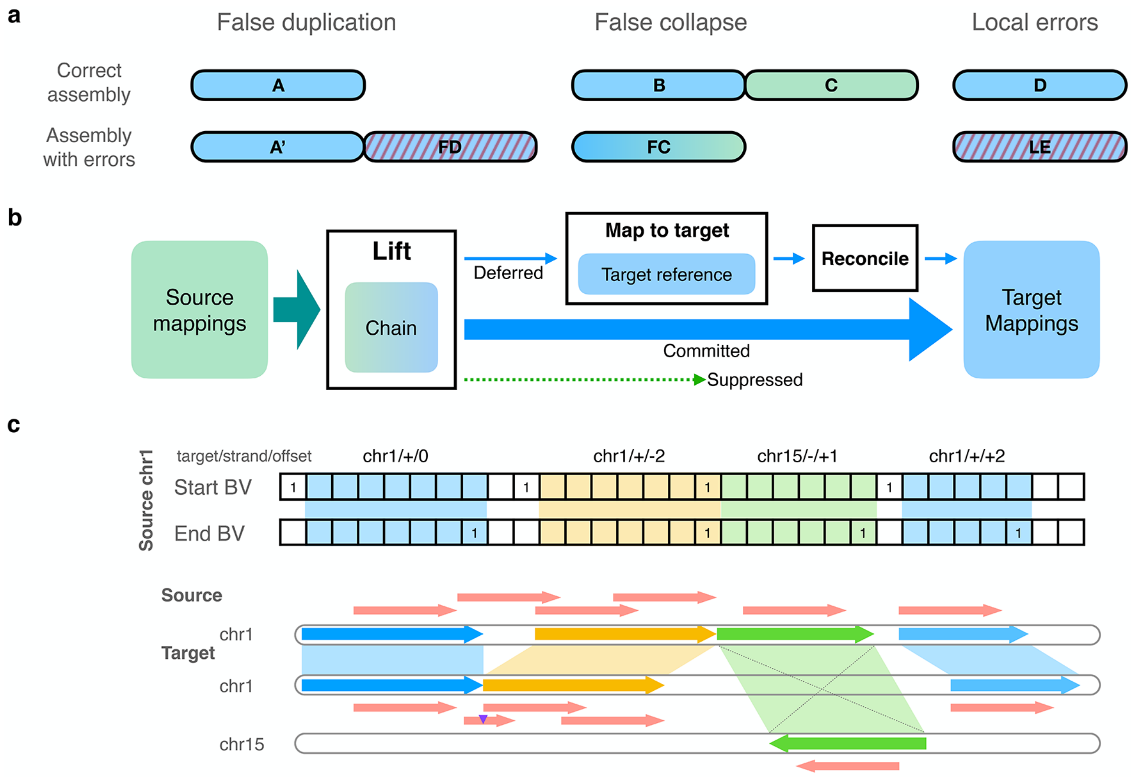


Figure 1: Overview of levioSAM2.

a, Common assembly errors in legacy reference genomes include false duplications (FD), false collapses (FC), and local errors (LE). **b**, The levioSAM2 workflow; A chain file provides information to lift mappings. Mappings which cannot be confidently lifted are deferred and re-mapped. The final output is a set of mappings to the target reference after reconciling the deferred and committed mappings. Reads originating in duplications and local errors can benefit from the "commit" and "defer–reconcile" strategies, where mappings to the source are considered. False collapses can be resolved by the "suppress" strategy, which avoids spurious alignments in the collapsed reference by only including mappings from a single orthologous copy. **c**, LevioSAM2-lift uses a pair of succinct bit vector data structures to efficiently query chain segments. LevioSAM2 lifts aligned reads from source reference to target reference using a chain file and updates the alignment CIGAR. Blue, yellow, and green arrows represent chain segments. Red lines represent mapped sequences. The purple triangle represents a 2-bp insertion in the source reference which requires a CIGAR update.

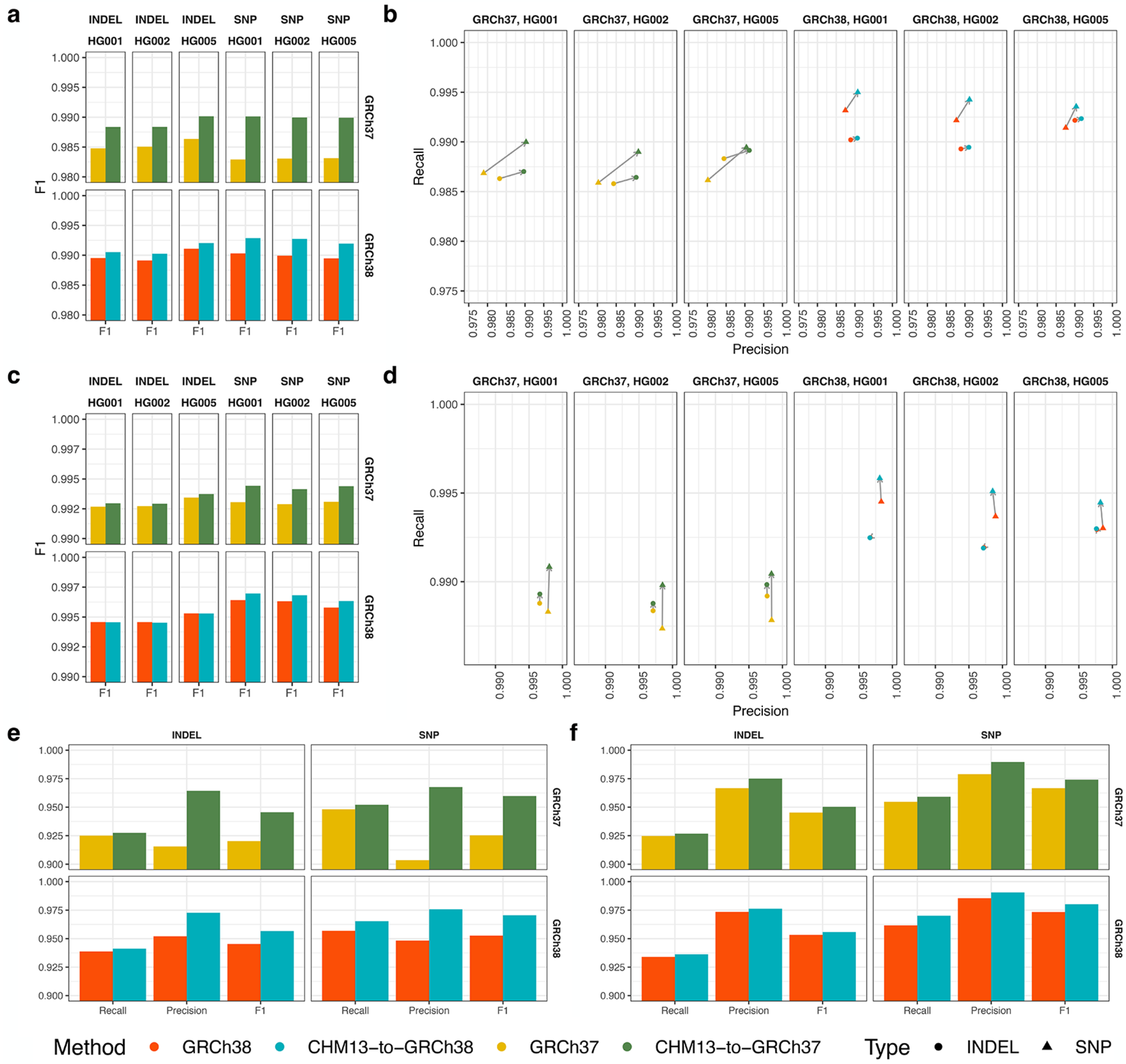


Figure 2: Small variant calling performance.

a, b, Variant calling accuracy using GATK-HaplotypeCaller for all GIAB v4.2.1 regions (**a**: F_1; **b**: recall and precision). **c, d**, Variant calling accuracy using DeepVariant for all GIAB v4.2.1 regions (**c**: F_1; **d**: recall and precision). **e**, Accuracy for challenging medically relevant genes (CMRG) for HG002 using GATK-HaplotypeCaller. **f**, Accuracy for challenging medically relevant genes (CMRG) for HG002 using DeepVariant.

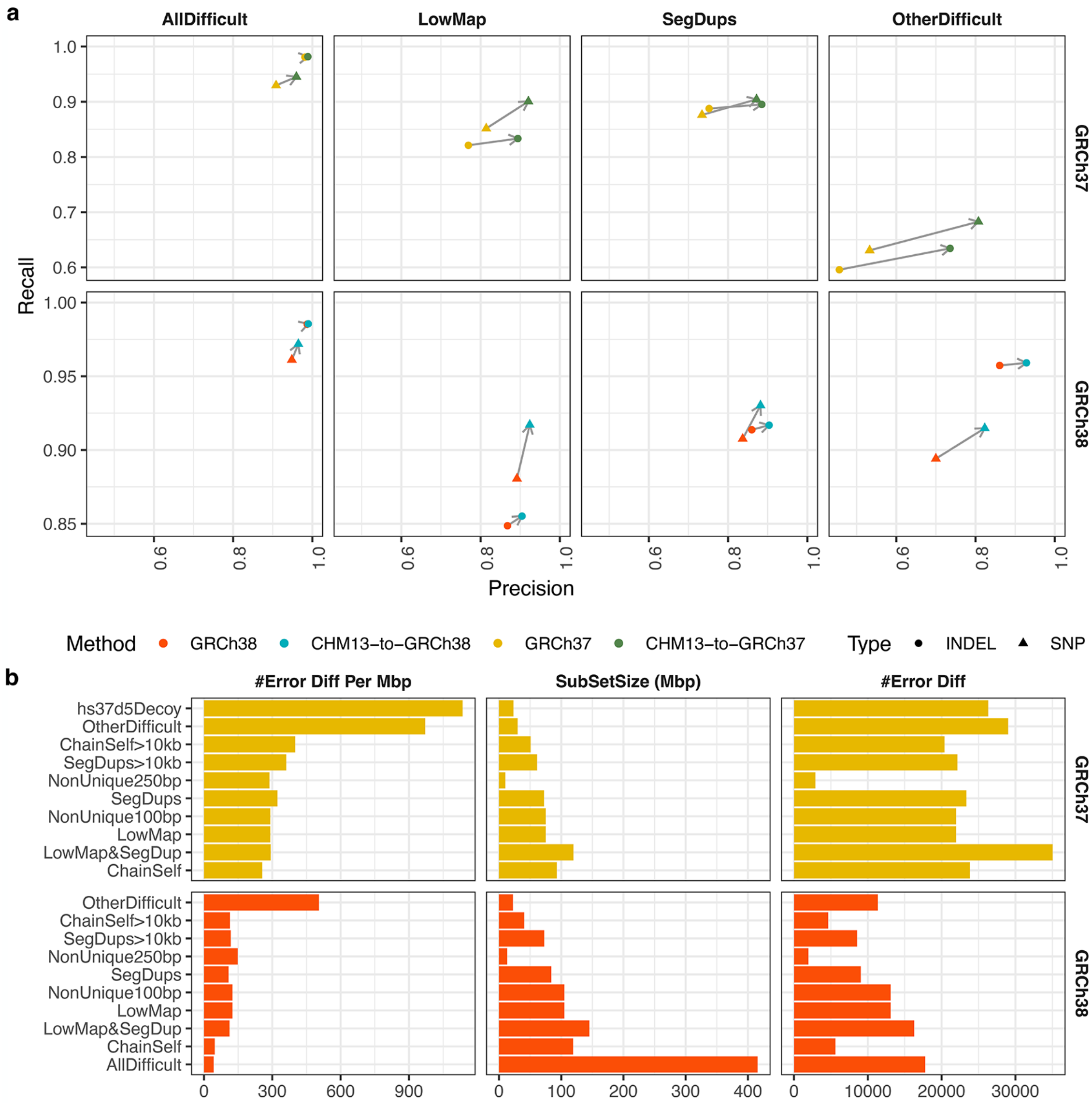


Figure 3: Small variant calling performance in difficult regions.

a, Small variant calling accuracy in major difficult genomic regions for HG002. We excluded difficult regions with low complexity or extreme GC content in the plot because all methods perform similarly. **b**, GIAB stratified regions with top small variant calling error reduction densities by levioSAM2. Small variants in both plots were called using using GATK-HaplotypeCaller.

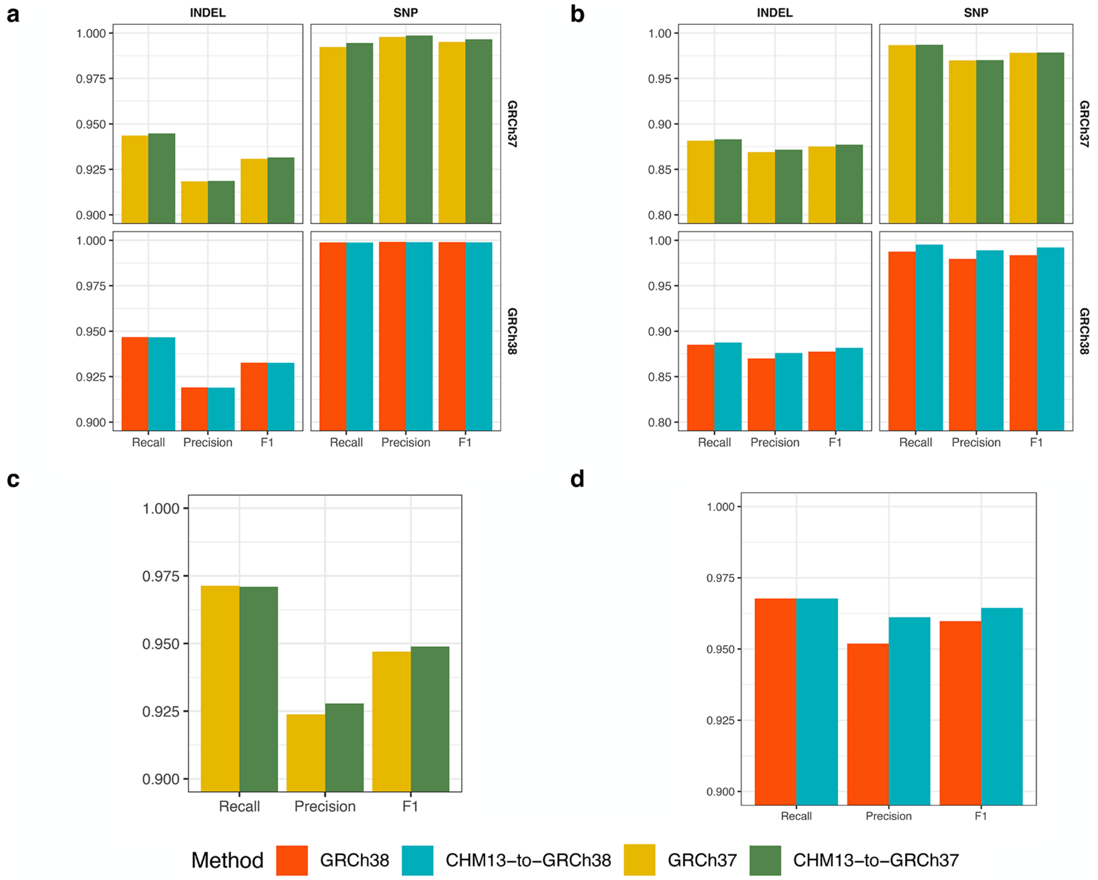


Figure 4: Small and structural variant calling using PacBio-HiFi reads from HG002.

a, b, Small variant calling using minimap2–DeepVariant in **a** GIAB v4.2.1 regions and **b** GIAB CMRG regions. **c**, Structural variant calling in GRCh37 GIAB Tier 1 benchmark regions. **d**, Structural variant calling in GRCh38 CMRG regions.

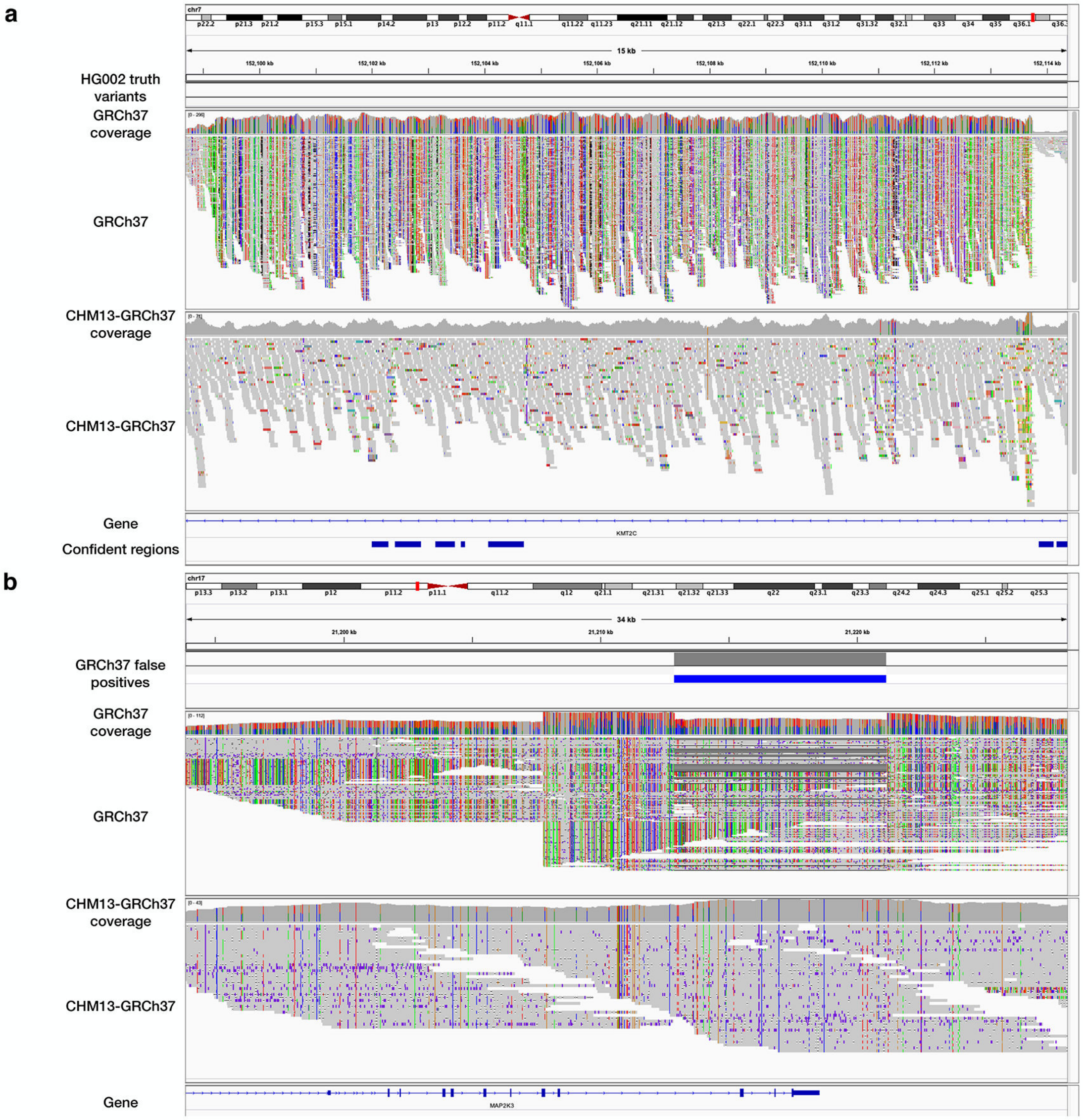


Figure 5: LevioSAM2 resolved large-scale mapping errors in medically relevant genes.

a, Mapping short reads to gene *KMT2C*. **b**, Mapping PacBio HiFi long reads to gene *MAP2K3*. The "GRCh37" track in both plots **a** and **b** show high mapping depth and high density of alternate alleles (bars with non-gray colors), suggesting collapsed mapping.

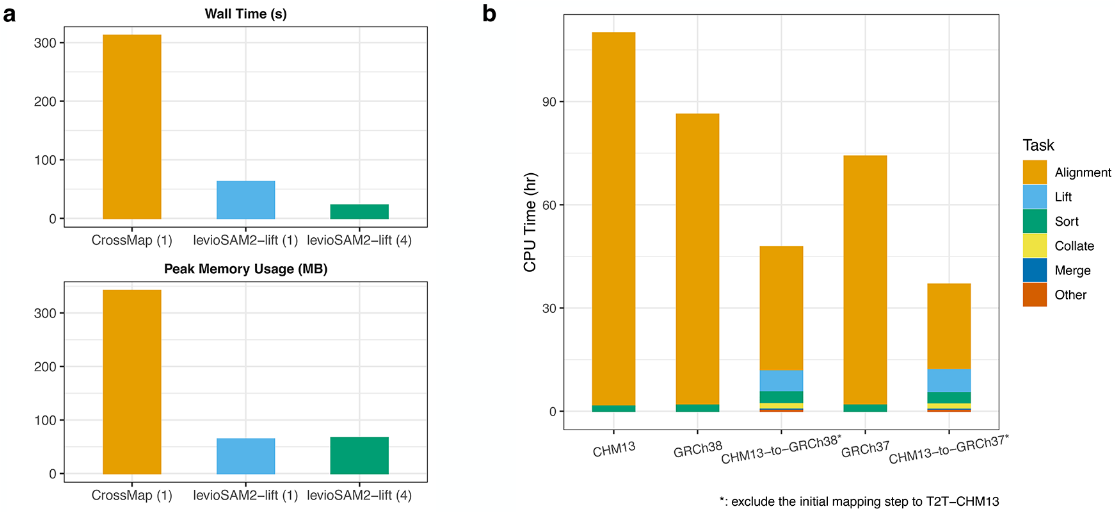


Figure 6: LevioSAM2 is computationally efficient and accurate when lifting from T2T-CHM13 compared to a direct mapping pipeline.

a, Computational efficiency of methods that support lift-over of alignments for a 0.3× paired-end WGS dataset. Numbers in parentheses show the number of threads used. **b**, CPU time usage of mapping using BWA-MEM vs. lifting over using levioSAM2 for a 30× paired-end WGS dataset. In the lift-over tasks, mapping to T2T-CHM13 was not included in the runtime measurement.

Table 1:

Real sequencing datasets and truth sets we used for evaluation. The Illumina data are from Google Health[32] and the PacBio data are from the Human Pangenome Reference Consortium[35]

Sample	Sequencing platform	Variant type	Truth variants	Coordinates
HG001	30× Illumina Novaseq	Small	GIAB v4.2.1[25]	GRCh37, GRCh38
HG002	30× Illumina Novaseq	Small	GIAB v4.2.1[25]	GRCh37, GRCh38
HG002	30× Illumina Novaseq	Small	GIAB CMRG[26]	GRCh37, GRCh38
HG002	28× PacBio HiFi	Small	GIAB v4.2.1[25]	GRCh37, GRCh38
HG002	28× PacBio HiFi	Small	GIAB CMRG[26]	GRCh37, GRCh38
HG002	28× PacBio HiFi	SV	GIAB SV Tier 1[28]	GRCh37
HG002	28× PacBio HiFi	SV	GIAB CMRG[26]	GRCh38
HG005	30× Illumina Novaseq	Small	GIAB v4.2.1[25]	GRCh37, GRCh38