nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-26865-w OPEN

Check for updates

Fast alignment and preprocessing of chromatin profiles with Chromap

Haowen Zhang[1,10], Li Song[2,3,10], Xiaotao Wang[4], Haoyu Cheng[2,5], Chenfei Wang[2,3], Clifford A. Meyer[2,3,6], Tao Liu[7], Ming Tang[2], Srinivas Aluru[1,8], Feng Yue[4,9], X. Shirley Liu[2,3,6]✉ & Heng Li[2,5]✉

As sequencing depth of chromatin studies continually grows deeper for sensitive profiling of regulatory elements or chromatin spatial structures, aligning and preprocessing of these sequencing data have become the bottleneck for analysis. Here we present Chromap, an ultrafast method for aligning and preprocessing high throughput chromatin profiles. Chromap is comparable to BWA-MEM and Bowtie2 in alignment accuracy and is over 10 times faster than traditional workflows on bulk ChIP-seq/Hi-C profiles and than 10x Genomics' Cell-Ranger v2.0.0 pipeline on single-cell ATAC-seq profiles.

[1] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA. [2] Department of Data Science, Dana-Farber Cancer Institute, Boston, MA, USA. [3] Harvard T.H. Chan School of Public Health, Boston, MA, USA. [4] Department of Biochemistry and Molecular Genetics, Feinberg School of Medicine, Northwestern University, Chicago, IL, USA. [5] Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. [6] Center for Functional Cancer Epigenetics, Dana-Farber Cancer Institute, Boston, MA, USA. [7] Department of Biostatistics and Bioinformatics, Roswell Park Comprehensive Cancer Center, Buffalo, NY, USA. [8] Institute for Data Engineering and Science, Georgia Institute of Technology, Atlanta, GA, USA. [9] Robert H. Lurie Comprehensive Cancer Center of Northwestern University, Chicago, IL, USA. [10] These authors contributed equally: Haowen Zhang, Li Song. ✉email: xsliu.res@gmail.com; hli@ds.dfci.harvard.edu

Chromatin profiling techniques, such as ChIP-seq[1], ATAC-seq[2], and Hi-C[3], have been widely used to study transcription factor binding[4], chromatin accessibility[5], and higher-order chromatin organization[6,7], respectively. Single-cell ATAC-seq (scATAC-seq) further enables the profiling of cis-regulatory elements in individual cells[8]. Standard analysis workflows, such as those used by the ENCODE project[9], start with read mapping by the popular short-read aligner BWA-MEM[10] or Bowtie2[11], along with alignment sorting and deduplication by SAMtools[12] and Picard[13]. These steps are the common bottlenecks, which may take hours or days to complete, compared to the downstream analysis steps such as peak calling by MACS2[14], which usually takes minutes. One reason for such inefficiency is that the comprehensive base-level alignment results for the purpose of variant calling are unnecessary for most chromatin biology studies. Furthermore, alignment filtering, deduplication, and other preprocessing steps are handled by different methods sequentially in a standard workflow, and each step requires parsing from compressed files. Such repeated I/O significantly increases the running time.

Our group previously developed minimap2[15], an efficient read aligner based on the minimizer sketch[16]. It was initially designed for long reads of high error rate and then extended for short accurate reads. Although a few times faster than FM-index-based short-read aligners such as BWA-MEM and Bowtie2, minimap2 more frequently misses short alignments that lack sufficient minimizer seeds. This becomes a severe issue in mapping scATAC-seq data when a large portion of the read sequence is used for barcoding and indexing, and the remaining genomic sequence in a read can be as short as 50 bp. Moreover, minimap2 has to slowly scrutinize the alignment to resolve the high sequencing error rate inherent in the long reads even when in the short-read mode, which could be unnecessary for the highly accurate Illumina short-read sequencing data.

In this study, we present an efficient read alignment and preprocessing method, named Chromap, based on the minimizer sketch (Fig. 1a). Chromap features a fast sorting-based procedure to generate mapping candidates and a fast alignment algorithm to pick the best candidate. To handle short reads better than minimap2, Chromap considers every minimizer hit and uses the read-pair information to rescue remaining missing alignments caused by the lack of low-frequency minimizers. Taking advantage of the observation that chromatin profiles are enriched only in a subset of the whole genome, Chromap caches the candidate read alignment locations in those regions to accelerate the alignment of future reads containing the same minimizers. Besides read mapping, Chromap also incorporates sequencing adapter trimming, duplicate removal, and scATAC-seq barcode correction, which further improves the processing efficiency (Methods). Chromap significantly reduces the computational time without losing accuracy.

Results

Performance on simulated data. We compared Chromap with other chromatin profiling aligners, namely BWA-MEM, Bowtie2, minimap2, STAR[17] (no-splicing mode) and Accel-Align[18] on three simulated whole-genome sequencing data sets with various read lengths (Fig. 1b). Except for STAR, the accuracy of these aligners was similar on the 100 bp and 150 bp paired-end data, about 98% for the five methods. On 50 bp paired-end data, BWA-MEM, Bowtie2 and Chromap had similar accuracy of around 96%, while minimap2, STAR and AccelAlign had worse performance at 94.1~95.7%. The comparison showed that Chromap achieved comparable alignment accuracy to BWA-MEM and Bowtie2 for a wide range of read lengths.

Performance on real ChIP-seq data. Next, we evaluated Chromap along with other aligners on real data sets, including ChIP-seq, Hi-C, and single-cell ATAC-seq data (Table S1). On a CTCF ChIP-seq data set from the ENCODE project, we first compared Chromap with BWA-MEM and Bowtie2. Among the 68 million fragments reported by any of the three methods (MAPQ ≥ 30), Chromap aligned 3% fewer fragments than BWA-MEM and 1.2% more than Bowtie2, and 99.8% of Chromap alignments were supported by either BWA-MEM or Bowtie2 (Fig. 1c). We next investigated the effects of the alignment methods on peaks called by MACS2 and included minimap2, STAR, Accel-Align in the evaluation. Peaks from Chromap alignment overlapped 99.8% with those from BWA-MEM and Bowtie2. While Chromap generated a comparable number of peaks as other methods, it created the fewest aligner-unique peaks (Fig. 1d, Fig. S1). Annotation of the peaks with ChIPseeker[19] did not find any aligner-specific bias in peaks from the alignment methods (Fig. S2). In addition, the differences of peak sets from the BWA-MEM, Bowtie2 and Chromap were significantly smaller than those between data replicates (Fig. S3). Notably, Chromap only took less than 5 min to complete the mapping, sorting, and deduplication process, while the second-fastest workflow based on Accel-Align, SAMTools and Picard required about 42 min. On the mapping step, Chromap (3.5 min) was 75% to 24.5 times faster than other alignment methods, supporting the efficiency improvement of Chromap (Table S2). We note that Chromap also reduced half an hour on the sorting and deduplication steps, confirming the advantage of integrating alignment and preprocessing in chromatin profiling analysis. Because minimap2, STAR and Accel-Align were not optimized for Hi-C data and could not be integrated to the single-cell data analysis pipeline, we excluded these methods in the following benchmarking.

Performance on Hi-C data. Chromap supports split-alignment, thus is compatible with Hi-C analysis. We compared the performance of Chromap and BWA-MEM on a Hi-C data set on the K562 cell line[7] by evaluating the downstream chromatin features such as chromatin compartments, topologically associating domains (TADs), and chromatin loops. The chromatin compartments (measured by the first eigenvector) and TADs (measured by the insulation score) called from the two aligners gave highly similar results, achieving Pearson correlation coefficients of 0.995 and 0.998 respectively (Fig. 2a, Fig. S4a). Although there is some divergence on the chromatin loops called by the two aligners, CTCF enrichment at the loop anchors supported these aligner-unique loops as genuine chromatin interaction loops (Fig. S4b, c. Methods). On this large data set with about 1.4 billion read fragments, Chromap spent 164 min to produce a processed alignment file in the pairs format[20] ready for downstream analysis. It was 13 times faster than a standard workflow with BWA-MEM and pairtools[20].

Performance on 10x Genomics scATAC-seq data. Last but not least, we tested Chromap on a 10k PBMC scATAC-seq data set from 10x Genomics with about 758 million reads and compared the results with CellRanger v1.2.0 and CellRanger v2.0.0, the official pipelines for processing scATAC-seq data developed by 10x Genomics based on BWA-MEM. Released in May 2021, CellRanger v2.0.0 substantially improves the computational efficiency over its predecessor along with other updates in preprocessing steps, such as deduplication criteria. We used all three methods for alignment and preprocessing followed by MAESTRO[21] for cell clustering and cell-type annotation (Fig. 2b). We evaluated the consistency of cell-type annotation using normalized mutual information (NMI), and found Chromap and

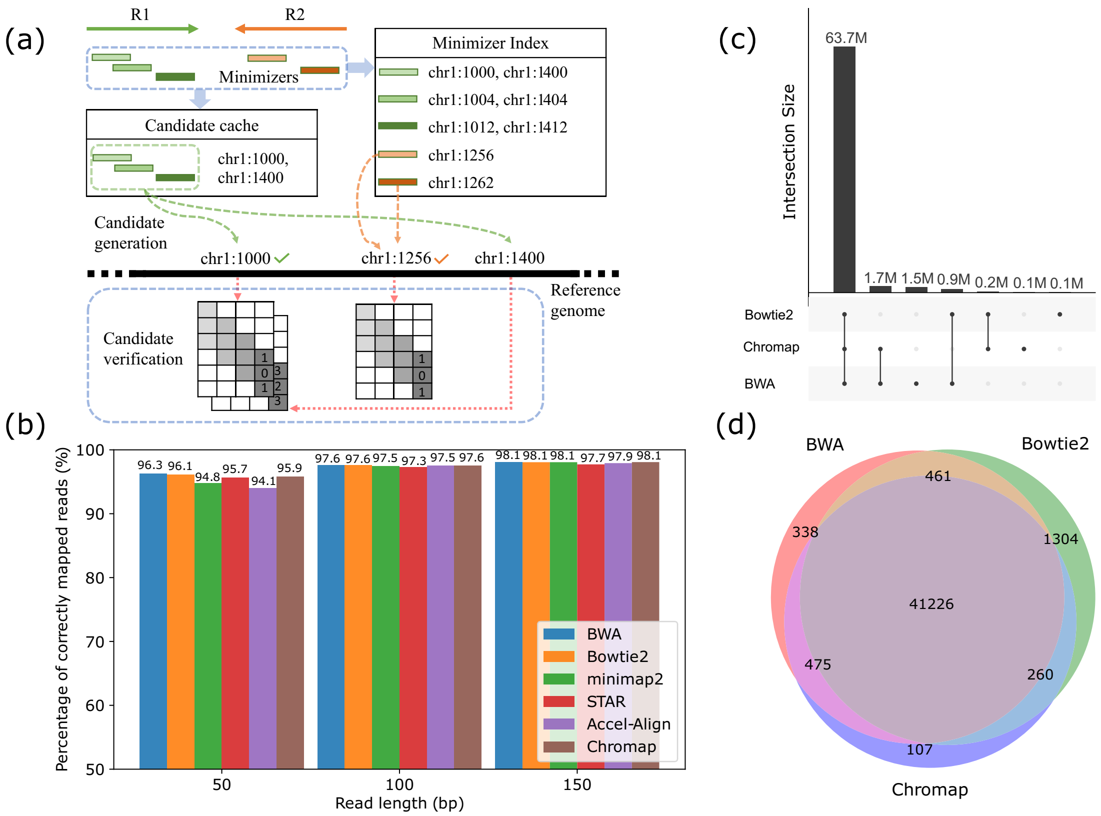

Fig. 1 Overview of Chromap. a Workflow of Chromap mapping a read-pair R1 and R2. First, their minimizers are extracted and then queried in the candidate cache and the minimizer index. The set of three minimizers of R1 is in the cache and the candidate mapping start positions are returned by the cache. The set of two minimizers in R2 is not in cache. So each of them is searched in the minimizer index and the occurrences of the minimizers are used to derive the candidate mapping positions. Then all the candidates are verified, which results in the final mapping. **b** Accuracy of methods on the simulated data with different read lengths. **c** Consensus of read alignments from Chromap, BWA-MEM, and Bowtie2 on bulk ChIP-seq data. **d** Overlapped peaks called from the alignments reported by different methods on bulk ChIP-seq data.

CellRanger v2.0.0 generated nearly identical results with NMI more than 0.96, higher than the NMI between the two CellRanger versions (Fig. 2b, Table S3, Methods). The lower consistency between CellRanger v1.2.0 and v2.0.0 suggested that alternating BWA-MEM and Chromap had less impact on the analysis than changing other preprocessing strategies. The clustering profiles were also highly similar between Chromap and CellRanger v2.0.0, no matter whether clustering was performed using the peak-based approach in MAESTRO or the bin-based approach in ArchR[22] (Table S4). On performance, Chromap generated the final alignment file in less than 30 min. It was 68 times faster than CellRanger v1.2.0 (33 h) and 16 times faster than CellRanger v2.0.0 (8 h). On this data set, Chromap directly obtained the candidates for about 120 million reads from the candidate cache of size two million entries which reduced the alignment time by 4%. The memory usage of Chromap is around 21GB, of which the candidate cache consumed about 1.7GB. Since the memory usage is dependent on the index file size, it is stable with respect to sequencing depth and regardless of applications to ChIP-seq, Hi-C, or scATAC-seq.

In summary, Chromap implements an efficient and accurate alignment and processing method for chromatin profiles. It is significantly faster than general-purpose aligners by taking full advantage of the nature of chromatin studies, i.e., read coordinate locations are more important for downstream analyses (Fig. 2c). Chromap further improves efficiency by integrating the adapter trimming, alignment deduplication, and barcode correction processing steps in the standard chromatin biology data workflows. With the decreasing cost of high throughput sequencing and increasing deeper sequencing coverage of chromatin profiles, Chromap will continue to expedite biological findings from chromatin studies in the future.

Methods

Overview of Chromap and improvements to minimap2. Though both Chromap and minimap2 build the minimizer index and extract minimizers from sequences as seeds to map the reads, they use distinct algorithms for seeding and for identifying alignment candidates. Minimap2 applies an expensive chaining procedure on the seeds to generate candidate mapping positions and then runs a slow dynamic programming algorithm that supports affine-gap penalty to verify those candidates. This complex procedure was initially designed for long reads and adapted for short reads later. It is overkilling and inefficient for short reads. Chromap, on the other hand, takes advantage of a light-weight candidate generation method, which is fast and sensitive to find candidate mapping positions for short reads. The candidates are supplemented using the read-pair information to improve mapping accuracy in repetitive regions, which minimap2 lacks. To verify alignment candidates, Chromap uses an efficient method to compute the edit distances of the read and its candidate mapping regions. Note that Chromap is not only an aligner like minimap2 but also an integrated tool that can preprocess the reads to remove adapters and correct the barcodes, and postprocess the mappings to remove duplicates. The details of Chromap are described below.

Index construction and query. Double-strand minimizers of reference genomes are collected and indexed using a hash table with minimizer sequences as keys and their sorted order of occurrences along with the reference as values (Fig. 1a). When mapping a read, Chromap retrieves the genome coordinates for each minimizer of the read. Due to the repetitive regions in the reference, some minimizers have high frequency, which can cause false-positive mappings and reduce mapping speed significantly. Thus by default, we mask minimizers occurring >500 times on the reference during query.

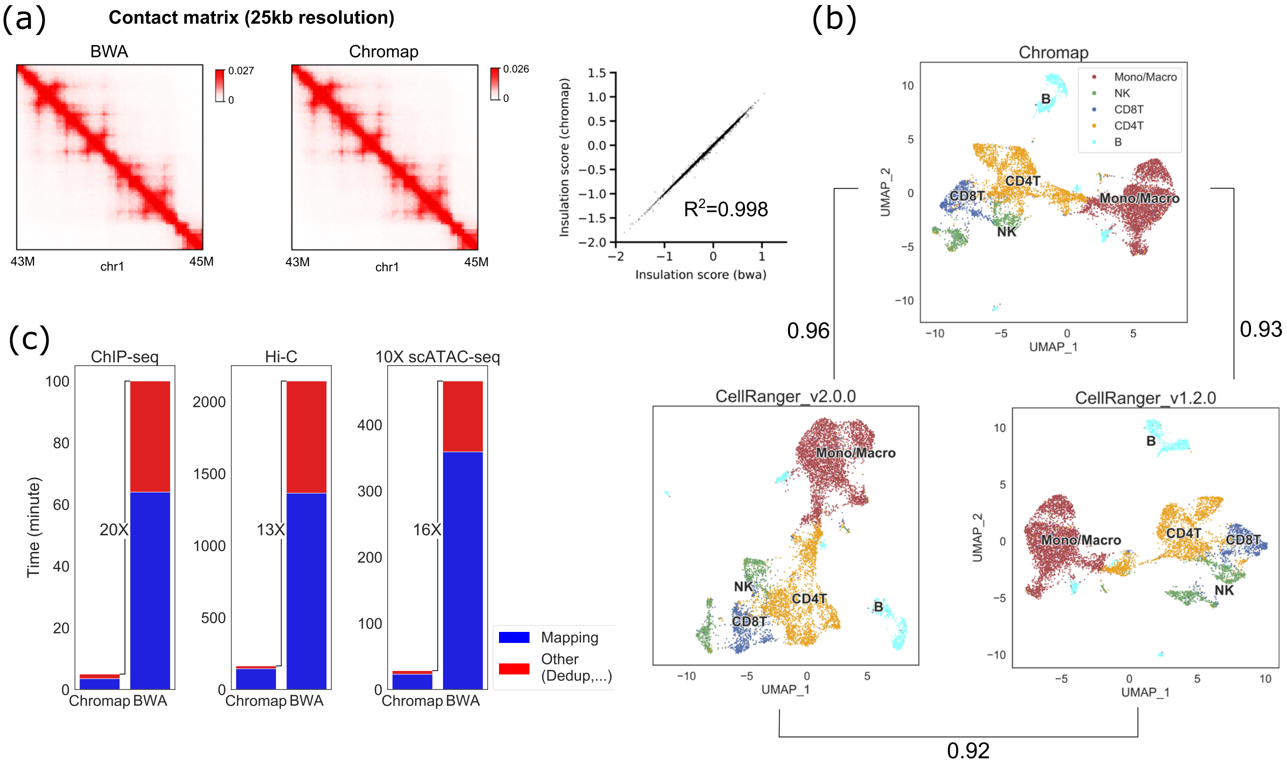

Fig. 2 Chromap on the large data set. a Comparison of Hi-C contact matrices at 25 kb resolution and insulation scores for TADs analysis derived from Chromap and BWA-MEM alignments. **b** Cluster annotation and NMI of the PBMC 10x Genomics scATAC-seq data based on the results of Chromap and CellRanger. **c** Running time of Chromap and workflows based on BWA-MEM on ChIP-seq, Hi-C, and 10x Genomics scATAC-seq data.

Adapter removal. For ATAC-seq or scATAC-seq, when a read contains the adapter sequence at the 3′-end, its fragment length can be shorter than the read length. To remove the adapters, for a pair of reads, if a prefix of one read has ≤1 Hamming distance compared with a suffix of the other read in the pair and the overlapped region is longer than a threshold l_{ovp}, we trim the bases outside the overlap (Fig. S5). We extract $l_{ovp}/2$ long seeds from one read, find the hits of the seeds in the other reads and verify those hits. This algorithm accelerates the trimming step and still guarantees finding overlaps within Hamming distance of 1.

Candidate generation. We define candidates for a read to be possible mapping start locations on the reference genome, which are estimated by exact minimizer hits (i.e., anchors) between the read and the reference. Formally, an anchor is a pair (x,y) where x denotes the minimizer start position on the reference and y denotes the minimizer start position on the read. Then the candidate can be estimated by this anchor as x-y. Co-linear anchors (i.e., chains) are a set of anchors that appear in ascending order in both the read and reference, which can be found by a dynamic programming algorithm[15] in quadratic time with respect to the number of anchors. While this algorithm can robustly identify chains for noisy long reads (>1000 bp with 5 ~ 10% error rate), we present a more efficient algorithm that can generate candidates for short reads with a low error rate. We generate candidates using all the anchors and then sort the candidates. During a linear scan on the sorted candidates, we merge the same candidates or candidates that have smaller than error threshold difference generated from multiple anchors. The error threshold is a user-defined parameter that constrains the edit distance between read and the genomic region. By allowing an error threshold in candidate merging, Chromap accommodates the insertions and deletions when generating the final candidates for a read. During the merging, Chromap records the multiplicity for each candidate, which is also the number of supporting anchors, and filters the candidates with fewer support than the user-defined threshold. For paired-end reads, chains were first generated for each end and then filtered by the fragment length constraint.

Candidate cache. Chromap stores the raw candidates in a cache for frequent reads to avoid repeated candidate generation for reads from peak regions. The cache is a hash table, where the key is a vector of minimizers and the value is the vector of candidates generated from the set of minimizers. The minimizers vector stores the M minimizers sequences m_i and the M-1 offsets between adjacent minimizers m_i. Chromap uses the function $h(m) = (m_1 + m_M) \bmod N$ to quickly map the vector to the $h(m)$-th entry in the hash table of size $N = 2{,}000{,}003$. The advantage of this mapping function is that the identical reads from both strands can access the same cached information. Furthermore, reads that are nearby in the genome have a greater likelihood of generating the same minimizer vector, and they can also share the same cache information.

Inspired by the count-min sketch[23], Chromap maintains a small count array of size $N' = 103$ in each cache entry to identify the most frequent minimizer vector from hundreds of different vectors mapped to the same cache entry, namely cache collidings. Chromap uses the function $f(m) = (m_1 \oplus m_M) \bmod N'$ to map the vector to the $f(m)$-th entry in the count table by computing the XOR of the minimizer codings, which has the same advantage of ignoring the read strand. Chromap then updates the cache table if and only if the count for the minimizer vector is more than 20% of the total count in the count array and is the dominant minimizer vector (show up more than half times) among the vectors mapped to the count array entry $f(m)$. As a result, Chromap not only stores in cache the candidates from frequent minimizer vectors, but also avoids unnecessary cache updates from the background noises.

Candidate supplement. Chromap supplements the candidates with read-pair information to recover the lost candidates due to the minimizer occurrence limit. For each read end, Chromap will pick its mate's candidate supported by the most number of anchors and use this mate's candidate as the estimation for the read coordinate. As a result, for each minimizer in the read end, instead of extracting all the occurrences on the reference, Chromap applies a binary search in the index entry to only select the occurrences within the range estimated read coordinate determined by the fragment size distribution. Chromap then executes the same candidate generation algorithm to supplement the candidates with the minimizer occurrences from the binary searches.

Candidate verification. Since each read can have multiple candidate mapping positions, we implemented a banded Myers' bit-parallel algorithm[24,25] to pick the optimal candidate coordinate with minimum edit distance to the reference genome. To further accelerate the verification step, we parallelized the algorithm using SIMD instructions on the CPU to align the read with multiple candidates on the reference simultaneously. We also modified the algorithm to efficiently trace back the alignment so that accurate start and end mapping positions can be obtained.

Split mapping. When the edit distance exceeds the threshold during the candidate verification step, we check if the length of the mapped read is greater than a certain length threshold. If the length of the mapping passes the length filter, the mapping is kept with an estimated mapping score as the mapped read length minus the edit distance. Note that for some of the Hi-C reads, there can be a small region (<20 bp), which cannot be mapped at the beginning of its 5′ end. To resolve this issue, when the mapping length is too short, the first 20 bp of the read is excluded and a second round of mapping of the remaining region is performed. If a mapping generated in this way passes the length filter, the mapping is then extended backward from its beginning to its maximum exact match. For paired-end data in split-alignment mode, Chromap ignores the constraints from the read-pair, such as the fragment length or strandness.

Deduplication. When the data set is small, all the mappings can be kept in the memory and sorted to remove duplicates. For large data sets or limited memory, we provide a low memory mode. It saves mappings in chunks temporarily on the disk and uses an external sort to merge them into the final mapping output in a low memory footprint. For scATAC-seq data, duplicates can be removed at either bulk level or cell level (default) based on the users' choice.

Barcode correction. Using the barcode whitelist provided by 10x Genomics, we correct barcodes that are not on the whitelist. Prior to the correction, the barcodes are converted to their bit representations and the abundance of each barcode is computed efficiently using a hash table. For barcodes outside the whitelist, all whitelisted barcodes within one Hamming distance from the barcode to correct are extracted by a set of efficient bit operations. Using the quality score of the mismatched base and the abundance of these whitelisted barcodes as a priori, we compute the posterior probability of correcting the observed barcode to the whitelisted barcodes. We make the correction if the highest probability of the observed barcode being a real barcode is > = 90%. The correction step is performed as part of the read mapping process which is in parallel of loading the next batch of reads.

Simulated and sequencing data for evaluation. In this work, we evaluated Chromap on various data sets including simulated whole-genome sequencing data, bulk ChIP-seq data, 10x Genomics scATAC-seq data, and Hi-C data (Table S1). One million fragments were simulated from the human reference genome GRCh38 using Mason[26] with an average sequencing error rate 0.1% and read lengths 50 bp, 100 bp, and 150 bp. The bulk CTCF ChIP-seq data on the human VCaP cell line were downloaded from ENCODE to test the tools on bulk sequencing data. The 10k PBMC scATAC-seq data set is publicly available from 10x Genomics and used to evaluate the performance of the tools on single-cell data. To investigate the impact of alternating BWA-MEM with Chromap on chromatin conformation analysis, we combined the two Hi-C data replicates from a previous study[7].

Evaluating performance for simulated and ChIP-seq data. In this work, we compared Chromap with five state-of-the-art short-read aligners minimap2(v2.17), STAR (v2.7.9a), Accel-Align (GitHub commit code 7217a9f), BWA-MEM (v0.7.17), and Bowtie2 (v2.4.2). STAR is designed to align RNA-seq data which contains spliced alignments across introns, so we used the options "--alignIntronMax 1 --alignEndsType EndToEnd" to forbid spliced alignment. When testing on simulated data, we converted all the alignments into PAF format and used the paftools to calculate the accuracy of alignments. Using the bulk ChIP-seq data, we compared the consensus of alignments and peaks among the aligners after filtering the alignments with MAPQs less than 30 based on ENCODE protocol (7 for Accel-Align). Accel-Align computes MAPQs in a different way, so we compared the distribution of MAPQs from all the aligners in the simulated data and found MAPQ 7 in Accel-Align was highly similar to MAPQ 30 in other aligners. All the methods were tested in a multiprocessing environment with 8 threads. Accel-Align was tested with option "-x" for the fast alignment-free mode.

Evaluating performance for Hi-C data. We compared Chromap and the standard 4D Nucleome Hi-C processing pipeline, which is based on BWA-MEM and pairtools[20], on a large Hi-C data set on human cell line K562. We filtered the alignments with MAPQ 0, which follows the default parameter settings in pairtools. Due to complexity introduced by the ligation junction in a Hi-C experiment, direct comparison of alignment coordinates would underestimate the consistency between the methods. Therefore, we compared the contact maps derived from the alignments at various resolutions. We compared the overall distribution of chromatin contacts at 25 kb resolution by using the stratum-adjusted correlation coefficients (SCC); the chromatin compartments measured by the first eigenvector of the normalized contact matrices at 100 kb; the TAD boundary strength measured by the insulation score at 25 kb resolution; and the identified chromatin loops at 10 kb.

To confirm that the difference between Chromap alignments and BWA-MEM alignments was smaller than the difference between biological replicates, we computed SCCs by using a Python implementation of HiCRep (https://pypi.org/project/hicreppy/, v0.0.6)[27] between Chromap and BWA-MEM on the same replicate (Chromap R2 vs. BWA-MEM R2) or between two replicates (BWA-MEM R1 vs. BWA-MEM R2). Because the original two replicates have disparate sequencing depths (R1: 1,048,612,352 vs. R2: 317,616,493), we first down-sampled R1 to make it match the sequencing depth of R2. The resulting SCC between Chromap R2 and BWA-MEM R2 was 0.998, which was significantly higher than SCC between BWA-MEM R1 and BWA-MEM R2 (0.945). HiCRep was run at 25 kb resolution, and the smoothing factor and the maximum genomic distance were set to 5 and 2 Mb, respectively. For the following chromatin conformation analysis, we merged the alignment results from the two replicates.

Both compartments and TADs were estimated using cooltools (https://pypi.org/project/cooltools/, v0.3.2). For compartments, the eigenvalue decomposition was performed on the 100 kb intra-chromosomal contact maps, and the first eigenvector (PC1) was used to capture the "plaid" contact pattern. The original PC1 was oriented according to a K562 DNase-Seq track (ENCODE accession code: ENCFF338LXW) so that positive values correspond to active genomic regions and negative values correspond to inactive regions. The Pearson correlation of PC1 was 0.995 between Chromap and BWA (Fig. S3a). For TADs, genome-wide insulation scores (IS) were calculated at 25 kb with the window size setting to 1 Mb. The Pearson correlation of the IS scores was 0.998 between the results from Chromap and BWA-MEM (Fig. 2b). Finally, we identified chromatin loops using HiCCUPS at 10 kb (https://pypi.org/project/hicpeaks/, v0.3.4). Among the 9455 and 9950 loops identified from Chromap and BWA-MEM respectively, we found 8,385 of them were supported by both methods (Fig. S3b). Furthermore, we found loop anchor sites that were uniquely identified by Chromap or BWA-MEM had a similar enrichment of CTCF binding peaks (Fig. S3c), suggesting those aligner-specific anchors could be biologically meaningful.

Evaluating performance for scATAC-seq data. We conducted comprehensive evaluations between Chromap and CellRanger on the 10k PBMC 10x Genomics scATAC-seq data to show that the clustering results were not affected by replacing CellRanger with Chromap. We compared the consistency of the cell-type annotations or cell clusters using normalized mutual information (NMI) and adjusted rand index (ARI) calculated by the Python package scikit-learn. We first computed the baseline NMI and ARI between CellRanger v1.2.0 and CellRanger v2.0.0. Chromap vs CellRanger v2.0.0 achieved a higher consistency score than the baseline score, suggesting the results from Chromap were highly consistent with CellRanger and were more consistent than CellRanger version changes (Table S3). To confirm the difference in the consistency score was insignificant, we created two replicates of the data set by randomly sampling 95% of the read fragments in the data set and applied CellRanger v2.0.0 to process these two replicates. The cluster-level NMI between the two down-sampled replicates (0.888) was lower than the NMI of the clusters generated from CellRanger v2.0.0 and Chromap (0.932), supporting that the impact from alternating CellRanger and Chromap is small. In addition, we also applied a bin-based scATAC-seq analysis method ArchR on this data set to evaluate the difference in the clustering caused by using Chromap and two CellRanger versions. Similar to the results on MAESTRO, we found alternating CellRanger to Chromap had tiny effects on the clustering results generated by ArchR (Table S4). Though CellRanger v1.2.0 is slow, it is easier to modify, and we were able to adapt it to use Bowtie2 as the alignment method (CellRanger v1.2.0_Bt2). Therefore, we could examine the impact of alternating the alignment methods on cell clusters. In this case, we ran Chromap with bulk level deduplication (Chromap_bulkdedup) as the setting in CellRanger v1.2.0. The NMI and ARI scores among CellRanger v1.2.0, CellRanger v1.2.0_Bt2 and Chromap_bulkdedup are all high (NMI > 0.9, ARI > 0.88, Table S5), suggesting that alternating the alignment methods BWA-MEM, Bowtie2 and Chromap had little impact on scATAC-seq analysis. CellRanger is a pipeline including data analysis steps after alignment and preprocessing, we measured its running time until the last "WRITE_ATAC_BAM" step in the log file.

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

Data availability

The data that support this study are available from the corresponding authors upon reasonable request. ChIP-seq data with two replicates are available from ENCODE: ENCSR265ARE and also on GEO (GSE105403). 10x Genomics data is available at: https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_pbmc_10k. Hi-C data is available in the SRA repositories: SRR1658693, SRR1658694, SRR1658695, SRR1658696, SRR1658697, SRR1658698, SRR1658699, SRR1658700, SRR1658701, SRR1658702. Source data are provided with this paper.

Code availability

The code used in the evaluation is available at: https://github.com/haowenz/chromap_evaluation. Chromap source code is available at https://github.com/haowenz/chromap[28].

Received: 10 June 2021; Accepted: 27 October 2021;
Published online: 12 November 2021

References

1. Barski, A. et al. High-resolution profiling of histone methylations in the human genome. *Cell* **129**, 823–837 (2007).

2. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nat. Methods* **10**, 1213–1218 (2013).
3. Lieberman-Aiden, E. et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009).
4. Farnham, P. J. Insights from genomic profiling of transcription factors. *Nat. Rev. Genet.* **10**, 605–616 (2009).
5. Buenrostro, J. D., Wu, B., Chang, H. Y. & Greenleaf, W. J. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr. Protoc. Mol. Biol.* **109**, 21.29.1–21.29.9 (2015).
6. Dixon, J. R. et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**, 376–380 (2012).
7. Rao, S. S. P. et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014).
8. Buenrostro, J. D. et al. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* **523**, 486–490 (2015).
9. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57–74 (2012).
10. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
11. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).
12. Li, H. et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
13. Broad Institute. Picard toolkit. *Broad Institute, GitHub repository* (2019).
14. Zhang, Y. et al. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* **9**, R137 (2008).
15. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
16. Roberts, M., Hayes, W., Hunt, B. R., Mount, S. M. & Yorke, J. A. Reducing storage requirements for biological sequence comparison. *Bioinformatics* **20**, 3363–3369 (2004).
17. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
18. Yan, Y., Chaturvedi, N. & Appuswamy, R. Accel-Align: a fast sequence mapper and aligner based on the seed-embed-extend method. *BMC Bioinforma.* **22**, 257 (2021).
19. Yu, G., Wang, L.-G. & He, Q.-Y. ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. *Bioinformatics* **31**, 2382–2383 (2015).
20. Dekker, J. et al. The 4D nucleome project. *Nature* **549**, 219–226 (2017).
21. Wang, C. et al. Integrative analyses of single-cell transcriptome and regulome using MAESTRO. *Genome Biol.* **21**, 198 (2020).
22. Granja, J. M. et al. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nat. Genet.* **53**, 403–411 (2021).
23. Cormode, G. & Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *J. Algorithm Comput. Technol.* **55**, 58–75 (2005).
24. Myers, G. A fast bit-vector algorithm for approximate string matching based on dynamic programming. *CPM.* **46**, 1–13 (1998).
25. Šošic, M. & Šikic, M. Edlib: a C/C ++ library for fast, exact sequence alignment using edit distance. *Bioinformatics* **33**, 1394–1395 (2017).
26. Holtgrewe, M. Mason: a read simulator for second generation sequencing data. http://www.seqan.de/projects/mason/ (2010).
27. Yang, T. et al. HiCRep: assessing the reproducibility of Hi-C data using a stratum-adjusted correlation coefficient. *Genome Res.* **27**, 1939–1949 (2017).
28. Zhang, H., Song, L., Fast alignment and preprocessing of chromatin profiles with Chromap. haowenz/chromap, https://doi.org/10.5281/zenodo.5558091 (2021).

Acknowledgements

We acknowledge the following funding sources for supporting this work: Breast Cancer Foundation [BCRF-20-100 to XSL] and National Institute of Health [R01HG010040, R01HG011139, and U01CA226196 to HL].

Author contributions

H.Z., L.S., X.S.L. and H.L. conceived the project. H.Z., L.S., X.W., H.C. and H.L. designed the method. H.Z. and L.S. implemented the algorithm. H.Z., L.S., X.W., H.C., C.W., C.M., T.L. and M.T. analyzed data or evaluated the method. H.Z., L.S., X.W., S.A., F.Y., X.S.L. and H.L. wrote the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare the following competing interests. X.S.L. is a cofounder, board member, SAB member, and consultant of GV20 Oncotherapy and its subsidiaries; stockholder of BMY, TMO, WBA, ABT, ABBV, and JNJ; and received research funding from Takeda, Sanofi, Bristol Myers Squibb, and Novartis. H.L. is a consultant of Integrated DNA Technologies and on the SAB of Sentieon and Innozeen.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-26865-w.

Correspondence and requests for materials should be addressed to X. Shirley Liu or Heng Li.

Peer review information: *Nature Communications* thanks the anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

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