Zhang *et al. Genome Biology* (2025) 26:406
https://doi.org/10.1186/s13059-025-03881-3

Genome Biology

SOFTWARE Open Access

# HAlign-G: rapid and low-memory multiple-genome aligner for large-scale closely related genomes

Pinglu Zhang[1,2,3†], Tong Zhou[1,2†], Yanming Wei[2,4], Qinzhong Tian[1,2,3], Yixiao Zhai[1,2], Yizheng Wang[1,2], Quan Zou[1,2], Furong Tang[5,6*†] and Ximei Luo[1,2*†]

†Pinglu Zhang and Tong Zhou contributed equally to this work.

†Furong Tang and Ximei Luo contributed equally to this work.

*Correspondence:
Furong Tang
Furong.Tang@hotmail.com
Ximei Luo
luoximei@uestc.edu.cn

Full list of author information is available at the end of the article

**Abstract**

HAlign-G is a fast and memory-efficient tool for large-scale multiple genome alignment. Using BWT-FM-LIS with an optimized K-band algorithm and star alignment strategy, it supports intra-species (HAlign-G1) and cross-species (HAlign-G2) alignment. Benchmarks show superior accuracy, efficiency, and memory use compared with existing methods. HAlign-G1 excels in speed and quality for intra-species data for multiple sequence alignment, while HAlign-G2 offers higher accuracy and structural variant detection for multiple genome alignment. Both versions handle millions of SARS-CoV-2 genomes and thousands of human chromosomes, enabling reliable evolutionary studies and supporting the construction of more stable phylogenetic trees, while enhancing Progressive Cactus performance.

**Keywords** Multiple sequence alignment, Multiple genome alignment, Structural variants, Phylogenetics

## Background

Multiple sequence alignment (MSA) is a crucial task in the field of bioinformatics, aimed at aligning three or more sequences and inserting gaps to maximize residue conservation in each column. Multiple genome alignment (MGA) represents a specialized form of MSA, is primarily employed for comparing the genome sequences of diverse species or populations to uncover their differences and shared features. With the advancement of sequencing technologies, an increasing number of high-quality assembled genomes have been generated [1–4]. By statistically analyzing these data, we can reveal potential factors contributing to phenotypic differences among different species or populations. This holds significant relevance in precision medicine, genome-wide association studies, and evolutionary genomics research [5–10]. However, large-scale data now poses significant challenges to genomic analysis [11–14].

The challenge of genome alignment lies in its extremely high computational and memory demand. In addition to small-scale mutations, insertions, and deletions within the

human genome, there are abundant repetitive sequences and large-scale events such as duplications, rearrangements, and inversions. These factors significantly increase the complexity of large-scale genome alignment tasks [5, 15].

Presently, the most widely used tools for genome sequence alignment primarily focus on pairwise sequence comparisons and have found extensive application in genomics research. Many of these tools employ a "seed-and-extend" methodology [16], as exemplified BLAST [17]. In this approach, they initially identify short and gapless sequence matches, referred to as "seeds", between sequences. Subsequently, they use an extension algorithm, such as Smith-Waterman [18], to extend the match from both ends of the seed sequence until the alignment score falls below a predefined threshold.

Seeds can be categorized based on whether they require an exact, contiguous match or have a fixed length [12]. Tools like BLAT [19] and STELLAR [20] use k-mer as exact-match seeds, while LASTZ [21] (an updated version of BLASTZ) employs gapped seeds. LAST [22] utilizes suffix arrays to search for variable-length adaptive seeds, and MUMMER [23] and CHAOS [24] respectively use suffix trees and threaded trie, respectively, to rapidly identify all exact contiguous match seeds of a given minimum length. Generally, adaptive, inexact, or gapped seeds tend to demonstrate higher sensitivity. Approximate-match approaches are generally more versatile, making them suitable for a wide range of sequence alignment tasks. By contrast, exact-match strategies are particularly effective for the alignment of highly similar sequences and for the analysis of repetitive genomic regions [13, 25, 26]. Following extension, local alignments that are adjacent, sequential, and consistently oriented can be linked by chaining programs such as AXTCHAIN [27], producing larger alignments in the output.

The development of multiple sequence alignment tools for genomes came later compared to pairwise sequence alignment, primarily due to their significant computational complexity, which limited their practical application. Currently, widely used multiple genome alignment tools include M-GCAT [28], progressiveMauve [29] (referred to as PMauve), Mugsy [30], progressiveCactus [31], and FAME [32]. For progressive multiple sequence global alignment, tools such as Parsnp [33], MAVID [34], MLAGAN [35], SeqAn::T-Coffee [36], and PECAN [37] are available. DIALIGN [38] employs a greedy approach for multiple sequence global alignment, whereas FSA [39] is based on probabilistic statistical hidden Markov models.

Among these tools, progressive Cactus [31] (referred to as PCactus) is the most robust cross-genomic multiple sequence alignment software, capable of aligning approximately 600 amniote genomes. However, its drawback lies in its strong dependence on the quality of the guide tree. In the context of intra-population genomic multiple sequence alignment, Parsnp stands as the most powerful tool, capable of aligning thousands of microbial genomes. Nevertheless, it suffers from the "reference bias", meaning it can only identify regions shared among all sequences. This leads to a significant reduction in alignment quality as the number of sequences increases. Furthermore, Parsnp cannot handle data with low similarity or large length differences.

To address the considerable computational complexity associated with large-scale multiple genome alignments, we developed HAlign-G for multiple genome alignment of closely related species, with two versions: HAlign-G1 and HAlign-G2. Each is designed for different applications—HAlign-G1 is more suitable for within-species alignments at the subspecies level, while HAlign-G2 is tailored for cross-species alignments among

closely related species, such as primates, though with slower speed. Experiments demonstrate that HAlign-G outperforms existing MSA and MGA methods in both performance and alignment quality, identifying more structural variants (SV) and completing tasks that other tools cannot (e.g., aligning 5,000 human chromosome 1 sequences or five million SARS-CoV-2 sequences). Moreover, phylogenetic trees derived from HAlign-G alignments are of higher quality, and using them as guide trees further improves Progressive Cactus accuracy. These results highlight HAlign-G's value for large-scale multiple genome alignment.

## Results

HAlign-G is implemented in C++. It offers two operational modes, which are determined by the user's choice of result file types. Selecting the generation of FASTA-format result files triggers multiple sequence alignments, while opting for MAF-format result files initiates multiple genome alignments. HAlign-G is designed to be user-friendly, requiring only basic parameters, such as input and output paths, with well-defined default values for more complex parameters. Complete experimental data statistics are available in Additional file 1: Table S1. The usage of HAlign-G is in Additional file 1: Table S2. All experiments were conducted on a Linux server running Ubuntu 22.04.5 LTS with dual AMD EPYC 7763 CPUs (256 cores, 2.45 GHz), 1 TB RAM.

### Alignment quality evaluation metric

The MSA quality was then evaluated using several widely adopted benchmark metrics: the Sum-of-Pairs score [40], the Q score [41], and the Total Column (TC) score [41]. The SP score is obtained by summing pairwise comparisons across all sequences, where matches are scored as 1, mismatches as − 1, and gaps as − 2; in this scheme, a higher SP score reflects higher alignment accuracy. To enable fair comparisons across datasets of different lengths, the Scaled-SP score is applied, which normalizes the raw SP score by sequence length. The Q score, ranging from 0 to 1, measures the similarity between a tested alignment and a reference, with a value of 1 representing a perfect match. The TC score evaluates only the correctly aligned columns relative to a reference alignment, making it a stricter criterion than either SP or Q. Taken together, these complementary metrics provide a robust and comprehensive evaluation of multiple sequence alignment quality.

In MGA, the most appropriate evaluation metrics are precision, recall, and F1-score; however, these metrics must be based on benchmarks with established ground truth. In the field of MGA, there is a lack of authoritative benchmarks for evaluating alignment accuracy. Currently, the only available benchmark is Alignathon [42], which supports only cross-species alignment methods and provides simulated datasets for primates and mammals. To assess the quality of MGA produced by our method, we therefore propose a new metric, the M-score (Match-score), designed to address the limitations of existing measures such as SP, TC, and Q, which were developed for FASTA-format multiple sequence alignments. Unlike these measures, multiple genome alignments are block-based and further complicated by structural variations. In a MAF-format alignment, the first sequence of each block is taken as the reference. For each column within the block, if the reference and a non-reference sequence share the same base and neither contains a gap or the character "N," the corresponding genomic coordinates in both sequences

are marked as matched. Each site is counted at most once: once a position is successfully matched in any block, it is considered matched, and subsequent matches do not increase the count. Across all sequences, the total number of matched bases is denoted as $M$, and the sum of the effective lengths of all sequences is denoted as $L$. The M-score is then defined as:

$$\text{M-score} = \frac{M}{L}, \quad 0 \leq \text{M-score} \leqslant 1$$

When $M$ and $L$ consider only two genomes, it can also be used to represent the similarity between the two genomes. Like coverage metric, but M-score emphasizes correctly aligned sites. Although it does not directly measure SV detection accuracy, M-score reflects the extent of correctly aligned regions and provides a complementary perspective on multiple genome alignment quality.

#### Evaluation of multiple sequence alignment quality on simulated dataset

For the comparative experiments on MSA quality, we used INDELible [43] to simulate mitochondrial-like sequences with sequence similarities ranging from 70% to 99%, following the parameter settings reported in FMAlign2 [13]. Figure 1 presents a detailed comparison of HAlign-G1, HAlign-G2, HAlign4 [14], HAlign3 [44], MAFFT [45], MUSCLE [41], and ClustalO [46] on simulated datasets, evaluating performance and accuracy in terms of scaled SP score, Q score, TC score, as well as computational time and memory consumption across different sequence similarity levels. Since all of these are well-established multiple sequence alignment tools, the comparison provides convincing evidence.

As shown in Fig. 1, It can be observed that HAlign-G1 consistently achieves the highest alignment quality across all tested similarity levels, substantially outperforming the classical tools. At low similarities, where most existing methods struggle, HAlign-G1 maintains remarkable accuracy, achieving Q and TC scores far above those of competing approaches. Among the other methods, MAFFT and HAlign4 generally performs best, while MUSCLE and ClustalO exhibit the lowest quality. At the lowest similarity level of 70%, HAlign-G1 achieved a remarkable Q score of 0.843 and TC score of 0.291, whereas all other methods remained below 0.8 and 0.2, respectively. This demonstrates that HAlign-G1 maintains superior accuracy and stability under low-similarity conditions, clearly outperforming existing approaches. In contrast, HAlign-G2 showed alignment quality comparable to other methods across all similarity levels, indicating it is less suitable for MSA tasks.

In terms of computational efficiency, MUSCLE and ClustalO again perform the worst, particularly MUSCLE. For MAFFT, HAlign3, both alignment time and memory usage increase notably as sequence similarity decreases. In terms of runtime, HAlign-4 performed the best, while HAlign-4, HAlign-G1, and HAlign-G2 all demonstrated excellent performance in memory usage. In summary, HAlign-G1 not only sustains superior alignment accuracy but also maintains outstanding efficiency, with minimal growth in time and memory consumption even for challenging low-similarity datasets.

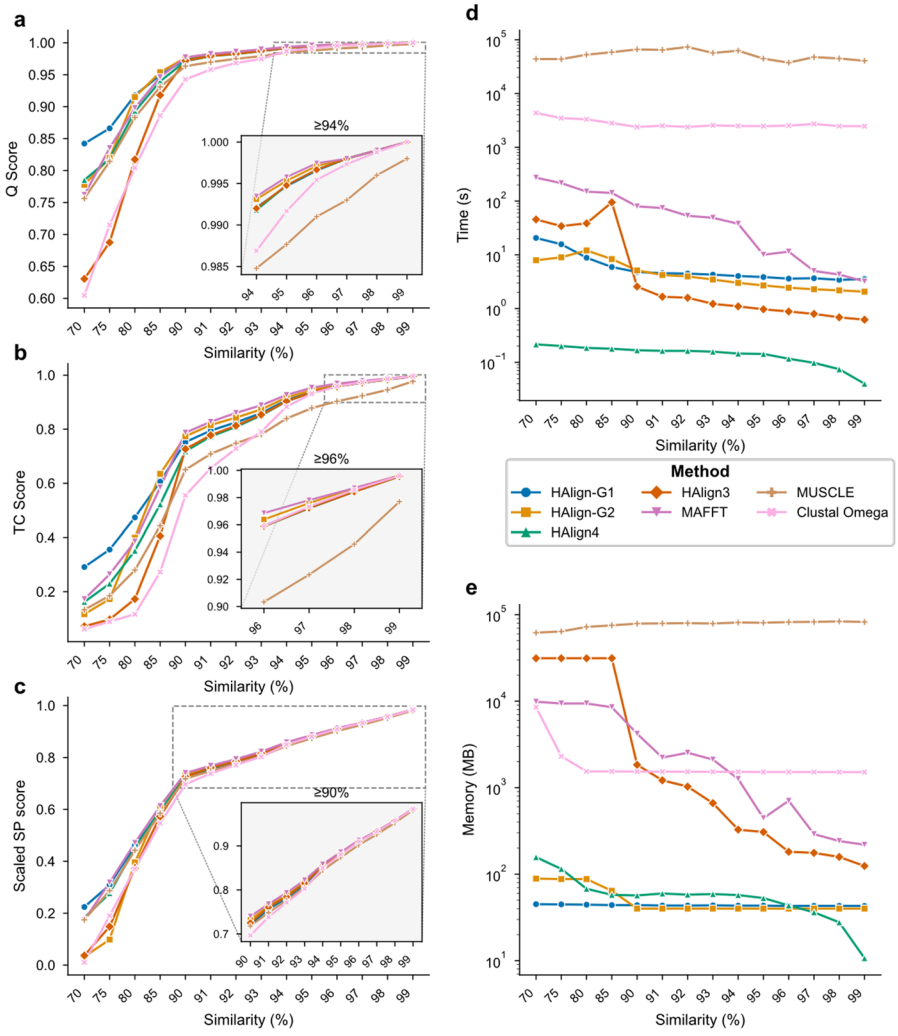


**Fig. 1** Comparison of MSA methods on simulated datasets across sequence similarity levels. **a** Q score. **b** TC score. **c** Scaled SP score. **d** Runtime.**e** Memory usage

### Evaluation of multiple genome alignment quality on simulated dataset

As shown in Table 1 and Additional file 1: Table S3, we compared the alignment accuracy of HAlign-G2 and Progressive Cactus on simulated primate and mammalian datasets of Alignathon [42]. All results, except those of HAlign-G2, were obtained using Progressive Cactus [31]. On primates, HAlign-G2 (HAlign-G1 is unable to perform cross-species alignments) demonstrated excellent performance, surpassing most existing methods and trailing Progressive Cactus by only 0.01. In terms of efficiency, Progressive Cactus required 33 min on 256 threads, whereas HAlign-G2 completed the task in just 6 min. These results highlight the strong advantage of HAlign-G2 in cross-species alignment among closely related genomes. In contrast, HAlign-G2 performed poorly on simulated datasets of distantly related mammals. This limitation arises because the star alignment strategy struggles with low-similarity species, and reference bias causes a substantial loss of alignments. Consequently, HAlign-G2 is best suited for closely related species, and we recommend its use for species with divergence times within approximately 20 million years. Despite its limited accuracy on distant lineages, its outstanding efficiency makes HAlign-G2 highly valuable in specific scenarios where performance is critical.

**Table 1** Precision, Recall, and F1 scores for simulated primates dataset from alignathon

| Aligner | Precision | Recall | F1 |
|---|---|---|---|
| Progressive Cactus | 0.986 | 0.991 | 0.989 |
| Cactus | 0.984 | 0.983 | 0.983 |
| VISTA-LAGAN | 0.978 | 0.983 | 0.980 |
| Mercator/Pecan | 0.940 | 0.996 | 0.967 |
| PSAR-Align | 0.980 | 0.995 | 0.988 |
| AutoMZ | 0.980 | 0.992 | 0.986 |
| TBA | 0.981 | 0.992 | 0.986 |
| Mugsy | 0.978 | 0.999 | 0.987 |
| progressiveMauve | 0.971 | 0.997 | 0.984 |
| Robusta | 0.941 | 0.986 | 0.963 |
| GenomeMatch (v1) | 0.898 | 0.997 | 0.945 |
| GenomeMatch (v2) | 0.898 | 0.972 | 0.934 |
| GenomeMatch (v3) | 0.905 | 0.261 | 0.405 |
| MULTIZ | 0.980 | 0.992 | 0.986 |
| HAlign-G2 | 0.971 | 0.988 | 0.979 |

To further and comprehensively evaluate the performance of HAlign-G in structural variation detection, we developed a simulator to generate eight types of intra-chromosomal structural variations: large insertions, large deletions, duplications, trans-duplications, inversion duplications, translocations, inversions, and trans-inversions. For the randomly generated reference sequence of approximately 4 Mb in length, the simulator randomly introduced two instances of each SV type, producing a new simulated sequence. The minimum SV length was set to 50 bp, and we considered 12 sizes (50–1000 bp at 100 bp intervals, plus 2000 bp). For each size, ten independent simulations were performed, yielding $12 \times 10$ datasets. We then evaluated HAlign-G1, HAlign-G2, Parsnp, Mugsy, and Progressive Cactus on the simulated dataset. Since pairwise alignments cannot detect large insertions or deletions, our evaluation focused on the remaining six SV types. Consequently, each simulated sequence contained 12 SVs (two of each of the six detectable types).

As shown in the Fig. 2a, HAlign-G2 achieved the highest M-score, followed closely by HAlign-G1, with only minor differences between HAlign-G and other methods, all exceeding 0.99. Consistent with this, Fig. 2b shows that HAlign-G2 also detected the largest number of structural variations, followed by HAlign-G1. Detection performance improved with increasing SV length: HAlign-G1 remained stable beyond 500 bp, while HAlign-G2 performed well above 100 bp. At 50 bp, only Parsnp detected SVs (fewer than two on average), indicating room for improvement in both versions of HAlign-G. Furthermore, as shown in Fig. 2c and d, HAlign-G2 delivered the best overall performance, with HAlign-G1 also performing strongly. By contrast, Progressive Cactus required the longest runtime, far exceeding other methods, while Parsnp consumed the most memory—nearly an order of magnitude higher than HAlign-G.

We further analyzed nine simulated datasets with structural variations of length 2000 (Fig. 3), evaluating a total of 108 sites ($9 \times 12$). Progressive Cactus identified 18 sites, Parsnp 19 sites, and Mugsy 26 sites. In sharp contrast, HAlign-G1 detected 102 sites, while HAlign-G2 detected 106 sites, missing only one site each in the second and sixth datasets. These results not only provide valuable guidance for selecting genome alignment software based on specific research needs but also underscore the markedly superior performance of HAlign-G in structural variation detection.

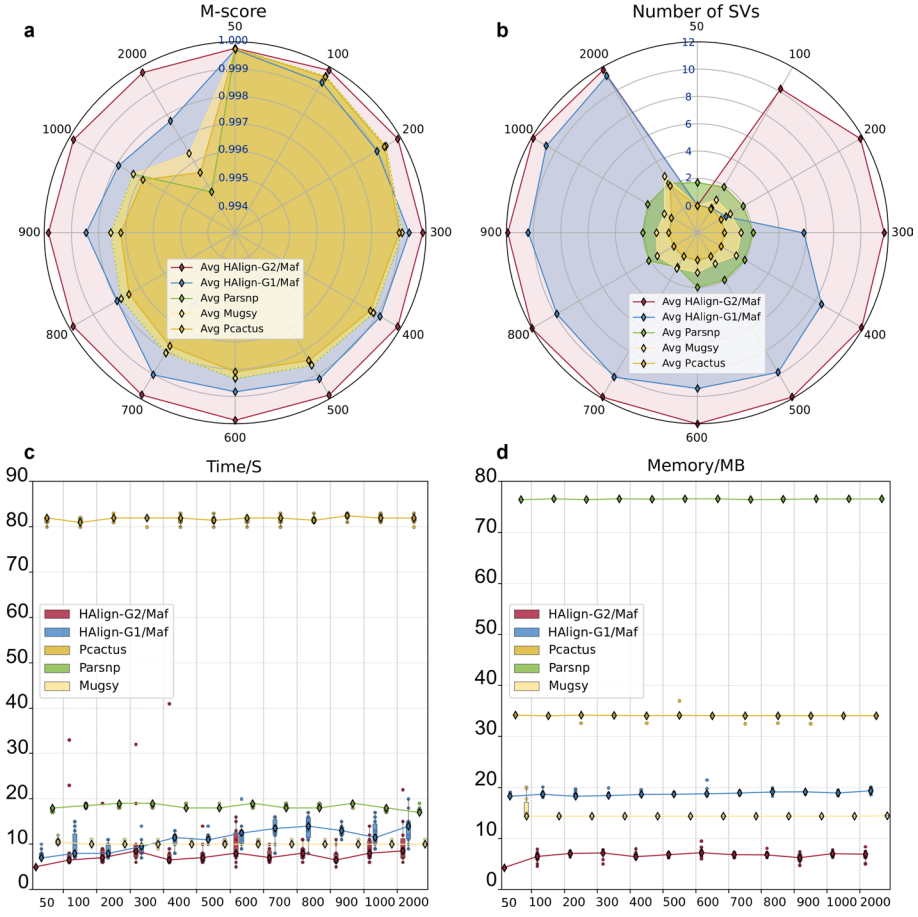


**Fig. 2** Pcactus, Parsnp, and Pmauve were compared with HAlign-G1 and HAlign-G2 on 12×10 sets of simulated structural variation datasets. **a** Average M-scores across different datasets. **b** Average number of structural variations detected. **c** Average runtime. **d** Average memory consumption

### Evaluation on small-scale genome datasets

We evaluated the FASTA and MAF modes of HAlign-G on five smaller genome datasets, and compared them with seven widely used multiple sequence alignment and multiple genome alignment software packages. These datasets include the mitochondrial genome, *Neisseria meningitidis*, *Streptococcus pneumoniae*, *Escherichia coli*, and three datasets of human chromosome 1. It is worth noting that the additional pairwise genome alignment tool MUMmer-q (with the delta-filter parameter set to q) was included solely for quality comparison with HAlign-G2. HAlign-G2/FASTA and HAlign-G1/FASTA were compared only with the previous version HAlign-3 at the multiple sequence alignment level, whereas HAlign-G2/MAF and HAlign-G1/MAF were compared with Progressive Cactus [31], Parsnp [47], Mugsy [30], KAlign [48], MLAGAN [35], and Progressive Mauve [29](see Additional file 1: Table S4 for usage of these softwares) at the multiple genome alignment level. During the evaluation, Parsnp was the only tool that successfully processed all five datasets, while the other six software packages encountered runtime crashes on certain datasets.

As shown in Fig. 4a and b, In the comparison of multiple genome alignment tools, HAlign-G2/MAF consistently maintained the highest quality and also outperformed MUMmer-q (The combination process of MUMmer4 with the delta-filter -q filtering mode), which adopts the same homologous region retrieval strategy. HAlign-G1/MAF,

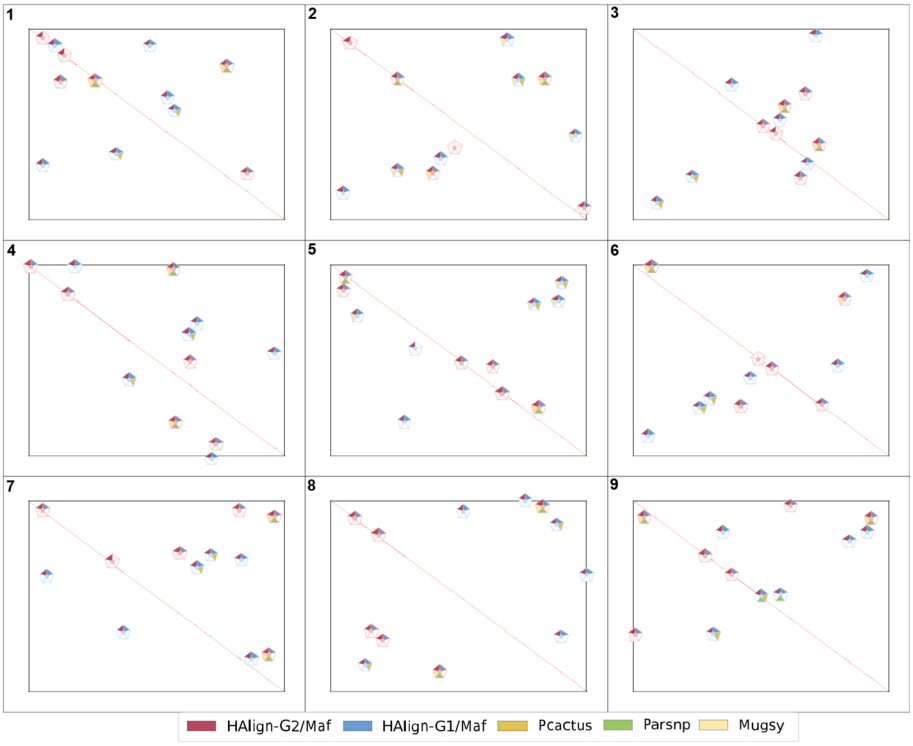


**Fig. 3** Comparison of structural variation detection among Pcactus, Parsnp, Pmauve, HAlign-G1, and HAlign-G2 on nine simulated datasets with a length of 2000 bp. In the figure, each pentagon represents a ground-truth structural variation site. Each pentagon is divided into five smaller triangles corresponding to the five software tools, and a triangle is highlighted if the corresponding tool successfully detected that site

however, still has room for improvement in terms of M-score on the Neisseria meningitidis and three Human-chromosome-1 datasets when compared with Mugsy and Progressive Cactus, while on the other three datasets it demonstrated quality close to that of HAlign-G2/MAF. In addition, both HAlign-G2/MAF and HAlign-G1/MAF exhibited low peak memory usage, with HAlign-G2/MAF showing a more pronounced memory advantage on longer sequences, whereas HAlign-G1/MAF generally achieved faster runtime.

In the comparison of multiple sequence alignment tools, the quality ranking was: HAlign-G2/FASTA > HAlign-G1/FASTA > HAlign-3. For memory consumption, HAlign-3 used the most, while HAlign-G2/FASTA showed a clear advantage when processing longer sequences. In terms of runtime, HAlign-G1/FASTA was faster, but the speed advantage of HAlign-3 diminished with increasing sequence length due to high memory usage, and in some cases it even failed to complete the task. Nevertheless, HAlign-3 still maintained an absolute speed advantage on highly similar short sequences, such as the mitochondrial genome.

On the Neisseria meningitidis and three human chromosome 1 datasets, HAlign-G2/MAF consistently achieved the highest accuracy, while HAlign-G1/MAF, though still among the top performers, showed room for improvement compared with Mugsy and Pcactus. Figure 4c showed that all four tools produced similar main alignment paths but differed in breakpoint regions. HAlign-G2/MAF detected the most structural variations outside the main path and aligned breakpoints more completely. HAlign-G1/MAF generated fuller but less precise breakpoints, whereas Mugsy and Pcactus produced paths

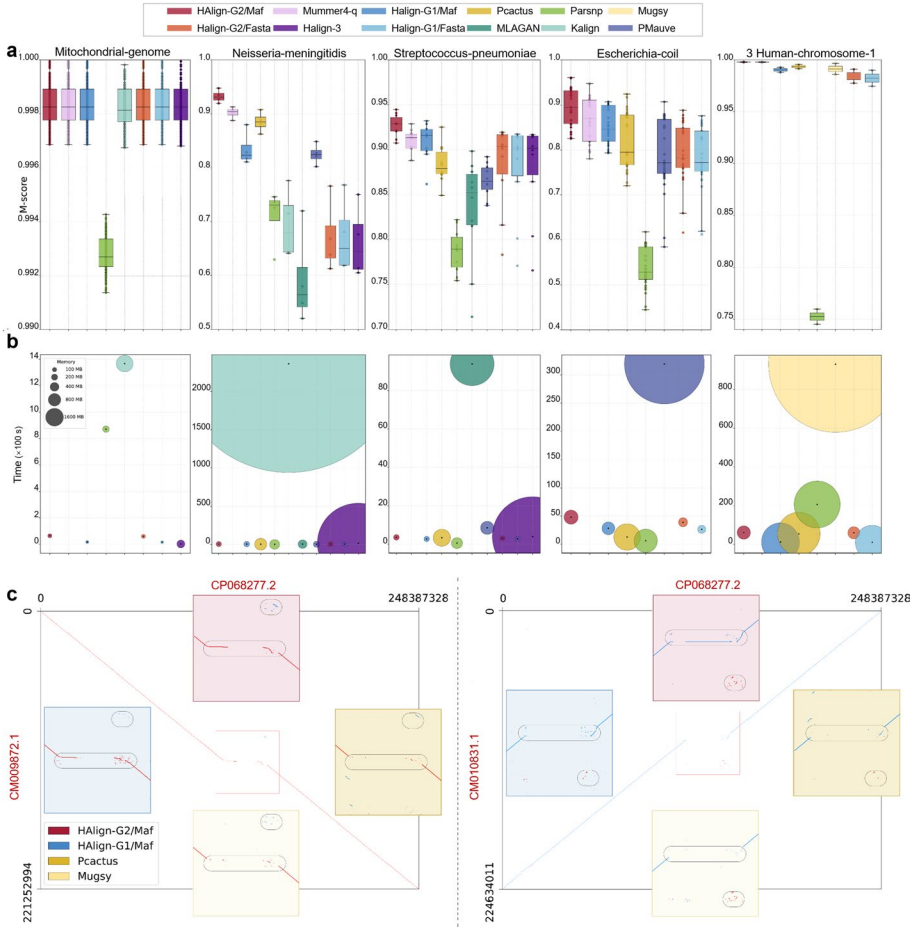


**Fig. 4** Comparative analysis of HAlign-G2, HAlign-G1, and other tools on five small genome alignments. **a** Comparison of M-scores across five datasets using different methods. **b** Comparison of runtime performance across five datasets using different methods. The horizontal position of the bubbles represents the running time, while the size of the bubbles indicates the memory size. **c** The alignment results of chromosome 1 between the human genomes CP068277.2 and CM010831.1 are shown. each point represents an aligned region, with red paths indicating forward alignments and blue paths representing alignments through reverse sequences. Points scattered outside the paths are potential structural variation loci identified by the respective software. The four smaller insets provide magnified views of the central portions of each software

that were overly sparse. Notably, HAlign-G1/MAF's stricter detection criteria mainly captured the overlapping subset of Mugsy and Pcactus, representing more reliable sites, but likely missed some lower-probability true variants, resulting in slightly reduced overall accuracy.

### Evaluation on large-scale genome datasets

To explore the performance of HAlign-G on large-scale datasets, we used a dataset of 18 human genomes (see Additional file 1: Table S5) to evaluate HAlign-G1 and HAlign-G2, comparing them with the only two existing software packages capable of handling such large-scale multiple genome alignment tasks: Progressive Cactus and Parsnp. In addition, to demonstrate the quality improvements of HAlign-G2 over MUMmer-q, which adopts the same homologous region retrieval strategy, we also tested this pairwise alignment tool using the M-score. In this evaluation, both HAlign-G and Parsnp were run

with 8 threads in parallel, whereas Progressive Cactus, by default, utilized all 80 available threads and required the guide tree generated by Parsnp as input.

As shown in Fig. 5a and b, HAlign-G2 consistently achieved the highest M-score and also outperformed MUMmer-q, despite the latter using the same homologous region retrieval method. Among the 24 chromosome, Parsnp and Progressive Cactus failed to produce results on four datasets ([13–15], and [22]). For the remaining 20 datasets, Progressive Cactus surpassed HAlign-G1 in only two cases (8 and 16), while HAlign-G1 achieved higher M-scores in the other 18. Compared with Parsnp, HAlign-G1 consistently outperformed it across all datasets. As shown in Fig. 5c, HAlign-G2 required the least memory, HAlign-G1 ranked second, and HAlign-G1 was the fastest. On the most complex dataset, chromosome 1, HAlign-G2 required 8.68 h and 12.44 GB peak memory, while HAlign-G1 required only 2.68 h and 12.80 GB peak memory. Such remarkably low computational resource requirements allow HAlign-G to perform these complex

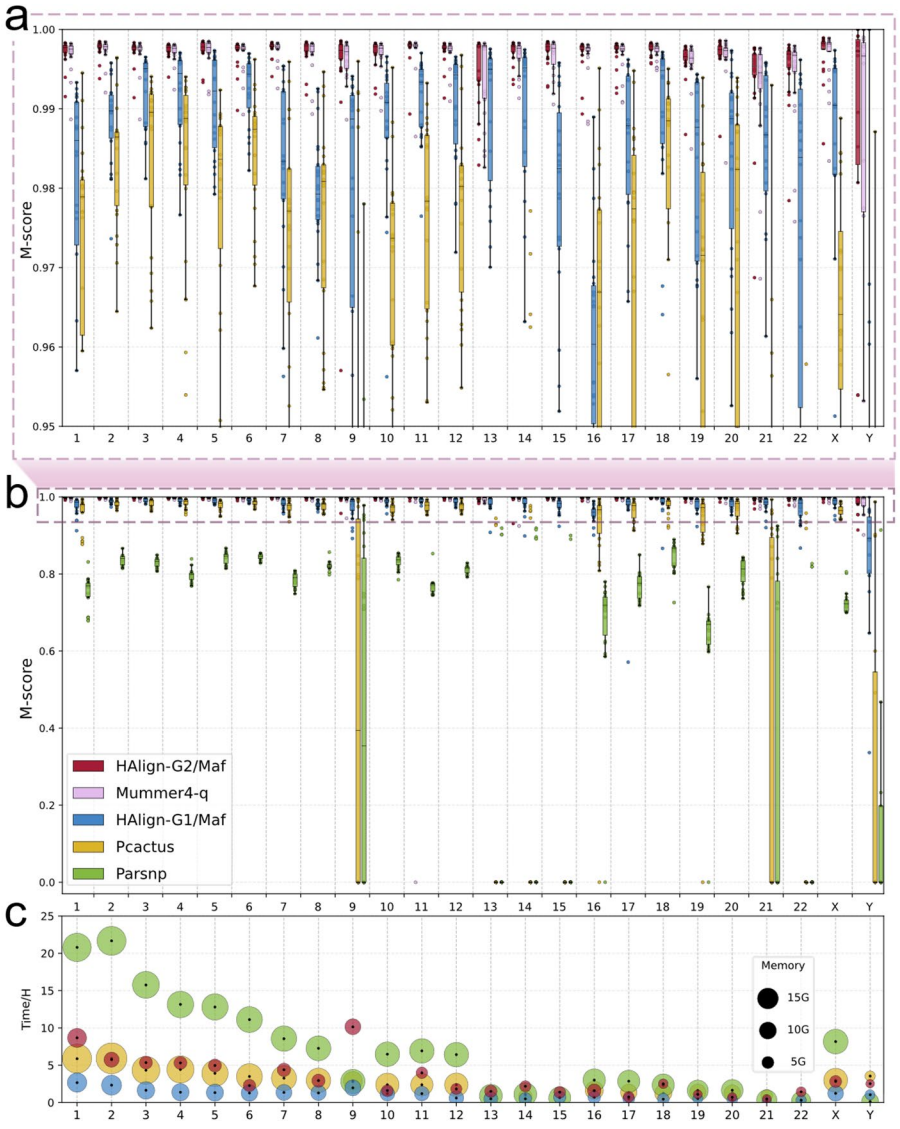


**Fig. 5** Comparative analysis of Pcactus, Parsnp, MUMmer-q, HAlign-G1, and HAlign-G2 on 24 chromosome datasets from 18 individuals. **a** M-scores of each method within the 0.95–1.0 range. **b** Overall comparison of M-scores across methods. **c** Performance comparison of the different methods

and demanding tasks even on ordinary personal computers, and greatly extend the scale of data it can handle on high-performance servers.

In addition, we applied HAlign-G in FASTA mode to align one million SARS-CoV-2 sequences. This experiment is of particular significance for large-scale sequence alignment, given both the enormous number and the high diversity of SARS-CoV-2 sequences. Leveraging HAlign-G's high throughput and strong multiple sequence alignment capabilities, we successfully completed this challenging task. As shown in Additional file 1: Figure S1, the average M-score achieved by HAlign-G1/FASTA was 0.99777, while HAlign-G2/FASTA attained an even higher score of 0.99784 (see Additional file 1: Figure S1). We also used HAlign4 to perform the alignment of the one million sequences and calculated the Scale SP score of the MSA. HAlign-G1 achieved a score of 0.51, HAlign-G2 achieved a score of 0.97, which was substantially higher than the − 0.19 obtained by HAlign-4, indicating that HAlign-G further improved alignment quality on large-scale SARS-CoV-2 data compared with HAlign-4.

We also applied both versions of HAlign-G to large-scale alignments involving 1,000 and 5,000 human chromosome 1 genomes (generated by duplicating the existing 24 human genomes). As shown in the Table 2, these datasets contained an exceptionally large number of sequences and highly complex alignments, yet HAlign-G delivered strong efficiency and accuracy. HAlign-G2 typically required longer runtimes than HAlign-G1, reflecting its emphasis on higher alignment quality. Generating MAF outputs increased memory demands for HAlign-G1, whereas HAlign-G2 proved more memory-efficient, even using less memory in its MAF experiments than in FASTA mode.

#### Quality of phylogenetic tree construction

Since Progressive Cactus is highly dependent on the quality of its guide tree, generating accurate trees is critical for genome alignment. To evaluate this, we constructed phylogenetic trees from the alignment results of HAlign-G1, HAlign-G2, and Parsnp using Caster, and supplied them as inputs to Progressive Cactus. Using the genomes of *Neisseria meningitidis*, *Streptococcus pneumoniae*, and *Escherichia coli*, we then compared the M-scores of the resulting alignments. As shown in Fig. 6a, Progressive Cactus consistently achieved higher alignment quality with guide trees derived from HAlign-G2, indicating that HAlign-G2 produces trees that more closely capture the true evolutionary relationships.

To evaluate phylogenetic tree quality, we used the benchmark dataset [49] comprising 101 *E. coli* genomes (51 ancestral and 50 sample sequences) with known evolutionary relationships. This dataset is widely used for benchmarking phylogenetic inference

**Table 2** Summary of HAlign-G performance on large-scale alignment tasks

| Data | Thread | Method | Time (h) | Memroy (GB) |
|---|---|---|---|---|
| 1000 Human-chromosome-1 | 8 | HAlign-G1/Fasta | 46.13 | 48.43 |
| | | HAlign-G2/Fasta | 505.39 | 39.97 |
| 1000 Human-chromosome-1 | 8 | HAlign-G1/Maf | 50.72 | 321.10 |
| | | HAlign-G2/Maf | 454.62 | 34.09 |
| 5000 Human-chromosome-1 | 64 | HAlign-G1/Fasta | 107.00 | 196.06 |
| | | HAlign-G2/Fasta | 505.00 | 195.87 |
| 1 M SARS-CoV-2 | 8 | HAlign-G1/Fasta | 8.71 | 87.77 |
| | | HAlign-G2/Fasta | 85.07 | 61.49 |

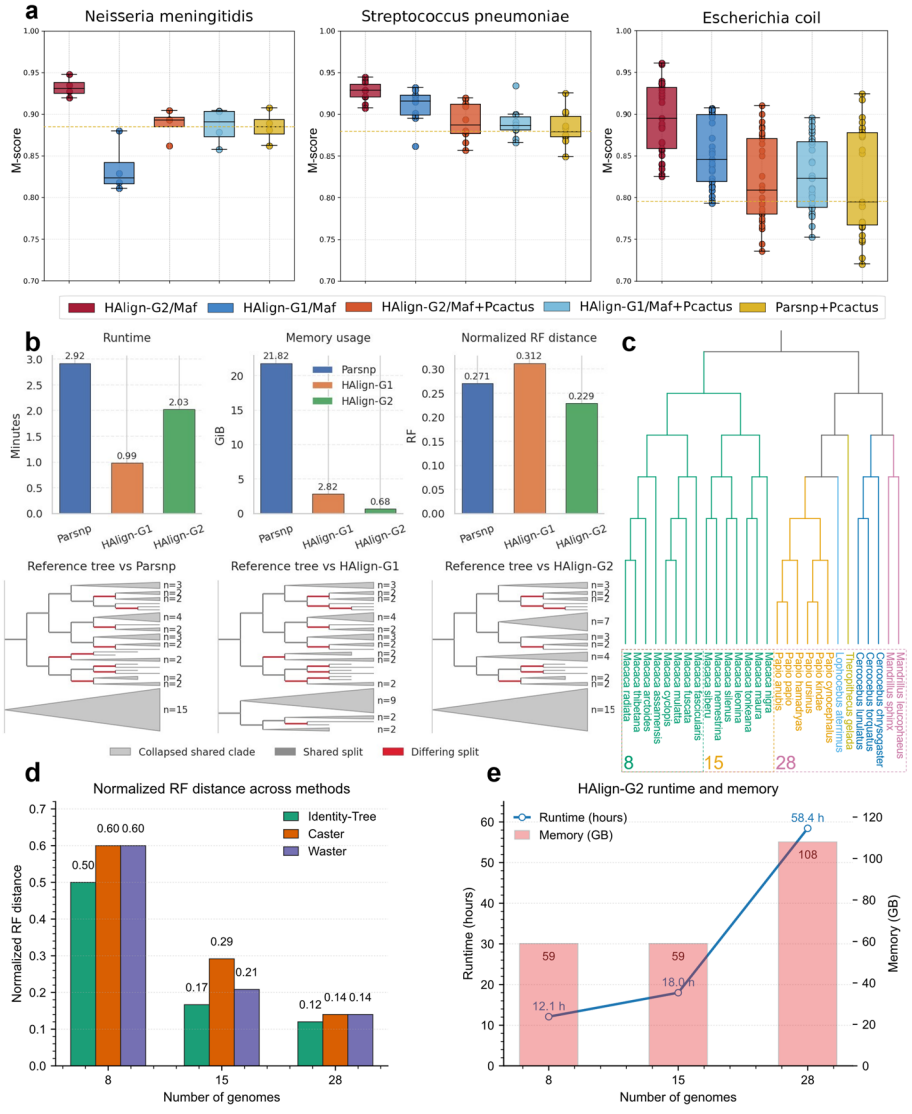


**Fig. 6** Comparison of HAlign-G and other methods in terms of phylogenetic tree reconstruction quality. **a** M-scores of Progressive Cactus alignments when using guide trees generated by different methods. **b** RF distances and performance comparison of phylogenetic trees generated by Parsnp, HAlign-G1, and HAlign-G2 on simulated datasets relative to the reference tree. **c** Reference phylogenetic tree of 28 primate genomes, along with the partitioning of these genomes into subsets of 8 and 15. **d** Normalized RF distances between phylogenetic trees constructed by three methods and the reference tree across varying numbers of primate genomes. **e** Runtime and memory usage of HAlign-G2 on primate genome datasets of different sizes

methods and provides a reliable ground truth for accuracy assessment. We performed alignments with HAlign-G1, HAlign-G2, and Parsnp using the same reference sequence, constructed phylogenetic trees using Caster [50, 51] based on these alignments. We then calculated their normalized Robinson–Foulds (RF) distances to the reference tree. As shown in Fig. 6b, HAlign-G2 achieved the lowest normalized RF distance (0.229), followed by Parsnp (0.271) and HAlign-G1 (0.312), indicating that HAlign-G2 can serve as an effective alternative to Parsnp for phylogenetic tree construction. The Fig. 6b also illustrates the specific branches where the three phylogenetic trees differ from the reference tree, showing that HAlign-G2 was able to correctly reconstruct a clade of size four, whereas Parsnp failed to do so. In addition, both HAlign-G1 and HAlign-G2 required

less runtime than Parsnp, and their peak memory consumption was nearly an order of magnitude lower.

Finally, to compare the quality of phylogenetic tree construction across species, We obtained 28 primate genomes [52] from Old World monkeys (Cercopithecidae), covering the genera *Macaca*, *Papio*, *Cercocebus*, *Mandrillus*, and their close relatives (see Additional file 1: Table S6). The phylogenetic tree of 28 primate genomes is shown in the Fig. 6c, from which we selected two subsets of 8 and 15 genomes, with the partitioning also illustrated in the Fig. 6c. We evaluated three approaches for tree construction. The first was based on HAlign-G2 alignments, where genomic similarity was calculated using the ratio *match/(match + mismatch)* and trees were built using the UPGMA algorithm; we refer to this method as Identity-Tree. The second approach also used HAlign-G2 alignments but employed Caster [50, 51] for tree construction. The third approach followed the official recommendation of Progressive Cactus, using WASTER [50, 53]. The reason Parsnp was not selected is that it does not support cross-species alignments involving multiple chromosomes.

Figure 6d shows that Identity-Tree achieved the smallest distance to the reference tree, most accurately reflecting the true evolutionary relationships, followed by WASTER and then Caster. These results indicate that phylogenetic trees derived from HAlign-G2 alignments can provide higher-quality guide trees for Progressive Cactus, thereby further enhancing its alignment accuracy. The relatively poorer performance of Caster may result from incompatibility between its algorithm and the outputs of HAlign-G2, or from uncertainties in the reliability of the reference tree itself. As these issues extend beyond the primary focus of this study on alignment performance, they are not explored further here. Figure 6e demonstrates that HAlign-G2 can complete alignments of up to 15 primate genomes within a single day, and can scale to 28 genomes in just 58 h, requiring only 59 GB and 109 GB of peak memory, respectively.

## Discussion

HAlign-G represents a parallelizable multiple sequence and multiple genome aligner specially crafted for extensive, lengthy sequences, including ultra-long genomic data. It possesses the capability to detect diverse structural variants within genome alignments. HAlign-G is a stable, efficient, and accurate tool that smoothly handles a wide range of data sizes and lengths. It typically outperforms other multiple sequence and multiple genome aligners in terms of speed and memory consumption while maintaining excellent accuracy.

To address the high time and space complexity of large-scale whole-genome multiple sequence alignment, we introduce an innovative approach called BWT-FM-LIS. This method identifies multiple anchor points within sequences to employ a divide-and-conquer strategy for handling ultra-long sequences, reducing time and space costs. Additionally, our efforts to enhance alignment performance and computational efficiency encompass the incorporation of techniques like affine gap penalties, path storage, and K-band constraints. These methods not only reduce computation time and storage space during dynamic programming but also ensure the consistency of alignment results with biological significance. To accurately identify structural variants, we have designed a scientifically rigorous structural variant identification module, allowing efficient integration of structural variations into global alignment results. We adopt a star alignment

integration strategy and leverage parallel computing techniques to better accommodate the processing requirements of large-scale datasets.

The results on simulated mitochondrial-like sequences demonstrated that HAlign-G1 substantially outperforms classical MSA methods under low similarity conditions, achieving higher Q and TC scores where most existing tools fail. This highlights its robustness in difficult alignment scenarios. On simulated primate and mammalian datasets from Alignathon, HAlign-G2 achieved accuracy comparable to Progressive Cactus but with dramatically lower runtime and memory usage. These findings indicate that HAlign-G provides an attractive balance between speed and precision, particularly for closely related genomes where reference bias is less pronounced.

Structural variation remains one of the most challenging aspects of genome alignment. Our experiments with simulated SV datasets revealed that HAlign-G2 achieved the highest M-scores and consistently identified more SVs than alternative methods. Notably, both HAlign-G1 and HAlign-G2 detected nearly all SVs in long (2000 bp) regions, whereas other aligners identified fewer than one-third of the sites. These results establish HAlign-G as a powerful tool for SV discovery, a capability that is critical for understanding genome evolution and disease-related genetic variation.

On small-scale real datasets, including bacterial genomes and human chromosome, HAlign-G2/MAF consistently achieved the highest alignment quality, while HAlign-G1/MAF offered a practical trade-off between accuracy and runtime. Importantly, both versions required significantly less memory than existing tools, some of which failed to complete certain datasets. On large-scale human genome datasets, HAlign-G2 maintained the highest M-scores across nearly all cases, while HAlign-G1 achieved superior runtimes. These findings emphasize that HAlign-G is the only current aligner capable of handling genome alignments at this scale on ordinary computing resources, significantly expanding its applicability in large genomic studies. The successful alignment of one million SARS-CoV-2 sequences further underscores its scalability and potential for rapid analyses in public health emergencies.

Phylogenetic reconstruction is another area where HAlign-G demonstrates significant promise. Our experiments showed that guide trees derived from HAlign-G alignments improved the alignment quality of Progressive Cactus, suggesting that HAlign-G captures true evolutionary relationships more accurately than Parsnp or Caster. Benchmark confirmed that HAlign-G2 achieved the lowest normalized Robinson–Foulds distances, establishing it as a reliable alternative for phylogenetic tree construction. Moreover, our analyses of 28 primate genomes revealed that Identity-Tree, built on HAlign-G alignments, provided the closest approximation to the reference tree, further validating its applicability in evolutionary studies.

Despite its strengths, HAlign-G has limitations. The star alignment strategy, while efficient, introduces reference bias that may lead to loss of alignments absent in the reference genome. This issue is particularly pronounced for distantly related species, where accuracy declines. We propose an iterative star alignment strategy as a promising future direction, in which multiple rounds of alignment ensure that all genome pairs are directly compared, thereby mitigating reference bias and enhancing robustness. Additionally, integrating advanced phylogenetic inference models could further improve the reliability of guide trees generated from HAlign-G alignments. Another valuable

direction is to enhance support for aligning ultra-long tandem repeat regions, such as centromeres, which remain challenging for current large-scale genome alignment [25].

## Conclusions

HAlign-G, a parallelizable multiple sequence and multiple genome aligner, excels in handling extensive, lengthy, and ultra-long genomic sequences, showcasing stability, efficiency, and accuracy. Its innovative BWT-FM-LIS approach, coupled with techniques like affine gap penalties and parallel computing, effectively addresses the challenges of large-scale whole-genome multiple sequence alignment, reducing time and space complexities. The incorporation of a scientifically rigorous structural variant identification module ensures the accurate detection and integration of structural variations into global alignment results. Notably, HAlign-G's stability and robustness surpass other aligners, as evidenced by its consistent execution across diverse datasets without the need for a guide tree. HAlign-G1 offers exceptional speed and memory efficiency, while HAlign-G2 emphasizes higher alignment accuracy and reliable cross-species support. Together, they provide flexible options to meet diverse research needs. Beyond alignment, HAlign-G's ability to produce phylogenetic trees that closely reflect true evolutionary relationships highlights its broader applicability to evolutionary and comparative genomics. In summary, HAlign-G emerges as a powerful tool for multiple sequence and multiple genome alignment, offering efficiency and accuracy for extensive genomic data. Its advancements in addressing alignment challenges, especially in ultra-large-scale, ultra-long data alignment, make it a valuable asset in genomics research, contributing significantly to the field's understanding of complex genomic structures.

## Methods

The HAlign-G workflow is shown in Fig. 7a. HAlign-G adopts a star-alignment workflow. It first preprocesses the input data and selects the longest sequence as the central reference. A BWT-FM index and hash dictionary are then built to support fast substring search. Using a divide-and-conquer BWT-FM-LIS strategy, anchor points are identified, and K-band dynamic programming [54] refines non-homologous regions. Structural variations are detected through dedicated modules and integrated into the results. Finally, all pairwise alignments are merged under the star strategy, producing FASTA (MSA) or MAF (MGA) output.

### Data preprocessing and central sequence determination

HAlign-G uses the FASTA file format as the input format. It supports two input scenarios: one involves providing a FASTA file containing multiple sequences, or providing a folder containing all the files. The input data in HAlign-G fully supports a wide range of degenerate bases and gap characters.

HAlign-G can identify whether the current sequence is an RNA or DNA sequence. To process data on an extremely large-scale data, HAlign-G converts all nucleic acid characters to uppercase and stores them in intermediate files. This approach facilitates later retrieval from external storage when needed, as opposed to keeping them in memory at all times. HAlign-G selects the longest sequence, with degenerate bases removed, which is considered to contain the most informative content. This sequence is then used as

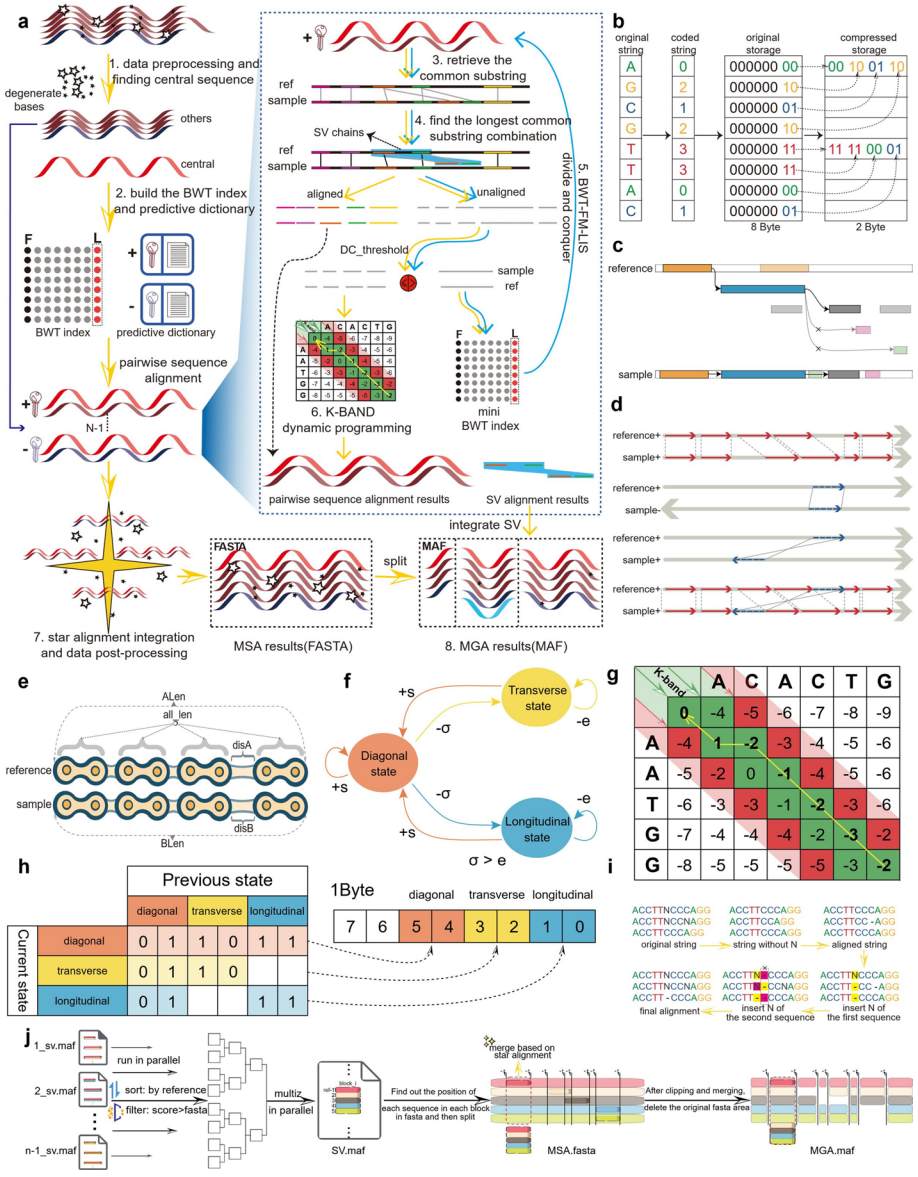


**Fig. 7** HAlign-G Workflow and Method Subflows. **a** Overall workflow of HAlign-G. The key difference in HAlign-G2 is that the retrieve the common substring process in HAlign-G1 is replaced by MUMmer4. **b** An example illustrates the compression of a DNA sequence to one-fourth of its original space. **c** The procedure identifies the primary chain within common substrings by filtering orange and gray segments based on relative position. When overlaps occur, the algorithm consistently retains the longest gray segment, discarding overlapping pink or green regions. The final primary chain is thus formed from dark orange, blue, and gray segments. **d** depicts the process of merging forward and reverse primary chains. **e** The process identifies structural variant chains by evaluating alignment lengths (ALen, BLen, all_len) and distances between adjacent substrings (disA, disB). Thresholds on length, distance, and relative differences ensure that the resulting chains represent reliable and biologically meaningful structural variations. **f** illustrates the concept of affine gap penalties. **g** provides a specific example of K-band dynamic programming with a length difference (diff) of 1. **h** depicts the compressed storage of the backtrack path table. **i** presents a specific example of deleting and inserting N in sequences. **j** delineates the process of integrating MAF result files concerning structural variants into the FASTA result files for multiple sequence alignments

the central sequence to accelerate the alignment process. Alternatively, users may also specify their own central sequence.

#### Building BWT index and prediction dictionary for the central sequence

HAlign-G utilizes the most used suffix array constructor, LibDivSufSort [55], developed by Yuta Mori, which enables the construction of a suffix array in linear time complexity. In multiple genome alignment, genome sequences from diverse species or individuals are often encountered, and these sequences may have varying orientations. HAlign-G employs MurmurHash3 [56] to create two separate hash dictionaries for the forward and reverse strands of the central sequence. Subsequently, these hash dictionaries are used for rapit searches using hash codes as keys. HAlign-G saves memory by using a BLAST-like binary compression method that encodes four bases per byte, reducing storage to one-quarter of the original and improving scalability (Fig. 7b).

#### Retrieving common substrings

HAlign-G utilizes k-mer to infer the alignment direction. It employs a Bloom Filter [57] to count the shared k-mers between each non-central sequence and the forward and reverse k-mer sets of the central sequence. By comparing the values of these two counts, the alignment orientation of the non-central sequence is determined, with alignment in the same direction as the central sequence corresponding to the forward strand, and alignment in the opposite direction corresponding to the reverse strand.

HAlign-G employs the FM-index on the BWT structure of the central sequence to identify common substrings shared with non-central sequences. Depending on the output format, the retrieval can be carried out in a single direction (FASTA) or in both directions (MAF) to detect potential inversion structural variations. Unlike forward retrieval, reverse retrieval advances by a larger step length after failures to quickly bypass low-probability regions. This strategy enhances efficiency while ensuring that both direct matches and structural variations are effectively detected.

#### Finding longest common substring combinations and identifying SV chains

The common substrings identified show high coverage, exhibit small positional offsets, and often overlap with each other. These common substrings include structural variations such as large-scale duplications, deletions, rearrangements, and inversions. Therefore, it is crucial to distinguish primary chains and structural variation chains. HAlign-G2 employs nucmer and delta-filter from MUMmer4 to perform the identification of the longest common substring, which is the only difference from HAlign-G1.

For small-scale genome data, where the number of common substrings found in the third step is limited, HAlign-G employs the most accurate but time-consuming dynamic programming algorithm to identify the primary chains. In the case of large-scale genome data, where the number of common substrings is abundant, running the $O\left(n^2\right)$ dynamic programming algorithm becomes extremely slow. Consequently, HAlign-G uses the following steps to find them (as shown in Fig. 7c):

(1) Initially, all one-to-many homologous intervals retrieved are refined to one-to-one mode based on the principle of relative positional proximity. This reduces the offsets between two sequences, ensuring cleaner subsequent alignments.

(2) To handle overlaps, the homologous common substring intervals are regarded as independent intervals and sorted by their right endpoints. A greedy algorithm is then applied to select the longest set of non-overlapping subintervals.
(3) Finally, the LIS algorithm is applied to resolve cross-ordering and obtain the final set of homologous intervals.

In HAlign-G's MAF mode, common substring searches are conducted in both forward and reverse directions. After obtaining common substrings in both directions, the task of finding primary chains must also be performed in both directions to obtain the longest common substring combinations. Among these, the combination of the longest common substrings in the same direction contains the most representative data. Subsequently, the longest common substring combination in the reverse direction mainly reflects several inversion structural variations. The reverse longest common substring combination is then divided into multiple inversion clusters, each corresponding to an inversion structural variation. These inversion clusters are merged with the forward primary chain. In case of overlap, HAlign-G trims the corresponding inversion clusters to obtain the true global primary chain (as shown in Fig. 7d). Once the inversion clusters are integrated into the global primary chain, forward non-homologous intervals corresponding to inversion clusters no longer require dynamic programming or divide-anfid-conquer alignment, thereby reducing the cost of aligning low-quality regions.

In addition to inversion clusters within the reverse primary chain, HAlign-G employs common substrings filtered out during the search for the forward and reverse primary chains as data sources for identifying other SV chains. Within this dataset, HAlign-G performs two separate searches to detect structural variations in both the forward and the combined inversion chains. To find these SV chains composed of compact and parallel common substrings, HAlign-G applies several thresholds to constrain potential common substrings within the chains (as illustrated in Fig. 7e).

After obtaining structured variation chains that are compact and orderly, these chains undergo a selection process. HAlign-G has defined a general selection rule after conducting tests on various multiple genome alignment programs. In the context of a two-genome alignment, the decision to retain a specific local alignment result (block) within the alignment result file is based on whether both the ref region and sample region in that block have been simultaneously covered by other blocks. If they have been covered by other blocks, the block is not retained. Finally, using K-band dynamic programming, HAlign-G fills in the alignments between common substrings in the chains to generate the complete structural variation alignment, which is then written in block format to the MAF file.

#### Divide-and-conquer BWT-FM-LIS

The BWT/FM-index can efficiently compress the central sequence and retrieve common substrings. Combined with a greedy algorithm and the longest increasing subsequence (LIS) algorithm, it selects the optimal combinations of common substrings. This process, termed BWT-FM-LIS, enables the rapid identification of similar fragments between two sequences.

When handling large genomes (e.g., the human genome), HAlign-G adopts Divide-and-Conquer strategy. It first applies a larger substring threshold [15] to quickly locate the main matching regions and construct a primary colinear alignment chain. Using this

chain, the two sequences are partitioned into separate segments. Within each segment, a smaller substring threshold [5] is then applied, and BWT-FM-LIS is repeated to further split the sequence fragments. This process continues until the resulting sequence fragments are sufficiently short (10000 bp) to be aligned precisely with dynamic programming.

#### Non-homologous region K-band dynamic programming

After the BWT-FM-LIS divide-and-conquer cycles, the alignment of non-homologous regions in the genome is reduced into many small dynamic programming tasks. HAlign-G optimizes dynamic programming in terms of scoring details, constraint ranges, and data storage.

An Affine gap penalty scheme is employed to distinguish the first and subsequent gap insertions: the penalty for the first gap insertion is higher than that for subsequent ones, ensuring that adjacent gaps are as close as possible in the alignment, thereby facilitating in obtaining more accurate variations and biological insights (see Fig. 7f). In our implementation, the gap opening penalty is set to d = 3, while the gap extension penalty is e = 1. This configuration discourages excessive initiation of gaps, yet permits the formation of longer contiguous gaps that better reflect biological insertions or deletions. All parameters, including match (+1) and mismatch (−2) scores, are user-configurable, allowing fine-tuning of the alignment process for different genomic datasets.

For highly similar sequences, HAlign-G reduces time and space complexity from $O\left(n^2\right)$ to linear $O\left(kn\right)$ by limiting the scope of dynamic programming. In this context, $n$ denotes the sequence length. HAlign-G employs the K-band method to confine dynamic programming to the vicinity of the diagonal. The justification for this approach is that the optimal alignment paths for two highly similar sequences are typically located near the diagonal. As a result, it is unnecessary to compute and fill the entire matrix. Instead, only the region near the diagonal requires computation. This region is referred to as the K-band (Fig. 7g). The baseline of the band is determined by the length difference between the two sequences, forming a region between two parallel lines near the diagonal. Subsequently, the band is extended to the upper right and lower left. The extension starts at 1 and increases by powers of 2 until the score of the current extension is no higher than that of the previous one. This process allows the dynamic programming to conclude.

The dynamic programming process for alignment typically requires storing multiple state matrices, particularly when affine gap penalties are applied, which results in high memory usage. HAlign-G overcomes this by introducing a compact path information table that records only the transition directions needed for backtracking (Fig. 7h). This approach avoids storing the full state tables and reduces memory consumption by nearly an order of magnitude, while maintaining alignment accuracy.

#### STAR alignment integration and data post-processing

In large-scale multiple sequence alignments of the human genome, aligning each sequence with the human reference genome as the central sequence is consistent with the STAR Alignment strategy. Each sequence only needs to be aligned to the human reference genome once. Afterward, following the principle of "once a gap, always a gap" the

gapped information from each pairwise alignments is combined to complete the multiple sequence alignment.

HAlign-G obtains gap information for each sequence through STAR alignment, but since degenerate bases (N) are removed during preprocessing, they must be reinserted in the final alignment. This process involves restoring the removed N bases, coordinating their insertion with gaps to avoid misalignment, and handling special cases (e.g., when multiple sequences contain N at the same position) to ensure the accuracy and consistency of the final alignment (see Fig. 7i).

#### Combining structural variants with MSA results

FASTA is the input format for HAlign-G and also one of its output formats. The MAF format is another output format of HAlign-G, which, in contrast to the FASTA format, can store alignments of structural variants such as inversions, translocations, and duplications.

HAlign-G has been modified to integrate and enhance the functionality of Multiz for merging two MAF files. HAlign-G performs sorting on MAF files before merging and saves the sorted result. Subsequently, it reads and performs Multiz merging operation on each block, writing the merged results (see Fig. 5-J). This approach does not keep the entire MAF file in memory and is better suited for handling large-scale genome data. Additionally, it is worth noting that Multiz only has a single-threaded merging capability, whereas HAlign-G needs to merge multiple MAF files for multiple genome alignment. Therefore, HAlign-G uses a layer-by-layer merging approach, where merging within each layer is independent. This allows for multi-threaded parallel processing.

In the final step of merging structural variant MAF files with multiple sequence alignment results in FASTA format, HAlign-G provides an efficient and lossless integration module. It first sorts the blocks in the structural variant MAF by the index of the central sequence and filters blocks with alignment quality exceeding that in the original FASTA. Then, it uses the start and end positions of each sequence in the block as cut points to split the FASTA alignment matrix into columns. Finally, it integrates the structural variant blocks from MAF with the corresponding blocks obtained by cutting the FASTA alignment based on STAR alignment. This process simultaneously removes structural variants from the corresponding regions in the original FASTA alignment, resulting in a final MAF file with integrated structural variants in the multiple genome alignment.

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-025-03881-3.

Additional file 1. Table S1. Overview of all benchmarking datasets, including sequence number, mean length (with/without degenerate bases), and total data size. Table S2. Command-line usage and parameter descriptions for HAlign-G, including default settings and user-configurable options. Table S3. Precision, recall, and F1 scores for multiple aligners on the Alignathon simulated mammalian dataset. Table S4. Software versions and exact execution commands for all compared alignment tools to ensure reproducibility. Table S5. Summary of the Complete Genomes of 18 Individuals. Table S6. Summary of 28 Primate Genomes. Figure S1. Distribution of M-scores comparing HAlign-G2/FASTA and HAlign-G1/FASTA on 1,000,000 SARS-CoV-2 sequences.

Additional file 2. Review history.

#### Acknowledgements

In addition to thanking HAlign-3 for its invaluable experience, which served as a solid foundation for the development and innovation of HAlign-G, we also express our gratitude to NCBI and GISAID for providing the necessary data support.

**Review history**
The review history is available as Additional File 2.

**Peer review information**
Veronique van den Berghe and Andrew Cosgrove were the primary editors of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

**Authors' contributions**
Pinglu Zhang, Tong Zhou, and Yanming Wei designed and implemented the method and constructed the datasets for method evaluation. Pinglu Zhang, Tong Zhou, Yixiao Zhai and Qinzhong Tian tested the software and wrote the article. Pinglu Zhang, Tong Zhou, Qinzhong Tian, and Yizheng Wang made and polished the figures. Quan Zou, Furong Tang, and Ximei Luo jointly conceived the study, guided experimental design, provided essential resources, and performed critical revisions of the manuscript. All authors read and approved the final manuscript.

**Funding**
The work was supported by the National Natural Science Foundation of China (No.62450002, No.62425107, No.62371347, No.62271353, No.62571375), Zhejiang Provincial Natural Science Foundation of China (No. LD24F020004), the Municipal Government of Quzhou (No.2024D001) and the Fellowship of China Postdoctoral Science Foundation (2023M731984, GZB20230365, 2024T170498). This work was also supported by Zhongguancun Academy projects 20240310 and 20240101.

**Data availability**
Code could be found at https://github.com/malabz/HAlign-G [58] under MIT License. All code and data in this study have been submitted to https://zenodo.org/records/17042774 [59]. The download links for the datasets used in this study are available at Additional file 1: Table S1, Table S5 and Table S6. The 1000 Human-chromosome-1 and 5000 Human-chromosome-1 datasets were produced by replicating the 18 copies of chromosome 1. Alignathon data can be found at https://cgl.gi.ucsc.edu/data/alignathon/ [42]. The benchmark dataset comprising 101 ***E. coli*** genomes can be found at https://cge.food.dtu.dk/services/evolution_data.php [49].

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

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

**Author details**
[1]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang 324000, China
[3]Zhongguancun Academy, Beijing 100094, China
[4]School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710126, China
[5]Quzhou People's Hospital, Quzhou Affiliated Hospital of Wenzhou Medical University, Quzhou, Zhejiang 324000, China
[6]State Key Laboratory of Molecular Oncology, Beijing Frontier Research Center for Biological Structure, School of Basic Medical Sciences, Tsinghua University, Beijing 100084, China

Received: 1 December 2023 / Accepted: 20 November 2025
Published online: 28 November 2025

**References**

1. Members C-N, Partners. Database resources of the National Genomics Data Center, China National Center for Bioinformation in 2025. Nucleic Acids Res. 2024;53(D1):D30–44.
2. Lewin HA, Robinson GE, Kress WJ, Baker WJ, Coddington J, Crandall KA, et al. Earth BioGenome Project: sequencing life for the future of life. Proc Natl Acad Sci U S A. 2018;115(17):4325–33.
3. Schoch CL, Ciufo S, Domrachev M, Hotton CL, Kannan S, Khovanskaya R, et al. Ncbi taxonomy: a comprehensive update on curation, resources and tools. Database. 2020;2020:baaa062.
4. Rhie A, McCarthy SA, Fedrigo O, Damas J, Formenti G, Koren S, et al. Towards complete and error-free genome assemblies of all vertebrate species. Nature. 2021;592(7856):737–46.
5. Armstrong J, Fiddes IT, Diekhans M, Paten B. Whole-genome alignment and comparative annotation. Annu Rev Anim Biosci. 2019;7(1):41–64.
6. Gibbs RA. The human genome project changed everything. Nat Rev Genet. 2020;21(10):575–6.
7. Logsdon GA, Vollger MR, Eichler EE. Long-read human genome sequencing and its applications. Nat Rev Genet. 2020;21(10):597–614.
8. Cui M, Liu Y, Yu X, Guo H, Jiang T, Wang Y, et al. MiniSNV: accurate and fast single nucleotide variant calling from nanopore sequencing data. Brief Bioinform. 2024;25(6):bbae473.

9. Zheng Z, Ren Y, Chen L, Wong AOK, Li S, Yu X, et al. Repun: an accurate small variant representation unification method for multiple sequencing platforms. Brief Bioinform. 2025;26(1):bbae613.
10. Wang Y, Zhai Y, Ding Y, Zou Q. SBSM-Pro: support bio-sequence machine for proteins. Sci China Inf Sci. 2024;67(11):212106.
11. Tian Q, Zhang P, Zhai Y, Wang Y, Zou Q. Application and comparison of machine learning and database-Based methods in taxonomic classification of High-Throughput sequencing data. Genome Biol Evol. 2024;16(5):evae102.
12. Kille B, Balaji A, Sedlazeck FJ, Nute M, Treangen TJ. Multiple genome alignment in the telomere-to-telomere assembly era. Genome Biol. 2022;23(1):182.
13. Zhang P, Liu H, Wei Y, Zhai Y, Tian Q, Zou Q. FMAlign2: a novel fast multiple nucleotide sequence alignment method for ultralong datasets. Bioinformatics. 2024;40(1):btae014.
14. Zhou T, Zhang P, Zou Q, Han W. HAlign 4: a new strategy for rapidly aligning millions of sequences. Bioinformatics. 2024;40(12).
15. Alkan C, Coe BP, Eichler EE. Genome structural variation discovery and genotyping. Nat Rev Genet. 2011;12(5):363–76.
16. Sahlin K, Baudeau T, Cazaux B, Marchet C. A survey of mapping algorithms in the long-reads era. Genome Biol. 2023;24(1):133.
17. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
18. Smith TF, Waterman MS, Fitch WM. Comparative biosequence metrics. J Mol Evol. 1981;18:38–46.
19. Kent WJ. BLAT–the BLAST-like alignment tool. Genome Res. 2002;12(4):656–64.
20. Kehr B, Weese D, Reinert K. STELLAR: fast and exact local alignments. BMC Bioinformatics. 2011;12(Suppl 9):S15.
21. Schwartz S, Kent WJ, Smit A, Zhang Z, Baertsch R, Hardison RC, et al. Human-mouse alignments with BLASTZ. Genome Res. 2003;13(1):103–7.
22. Kielbasa SM, Wan R, Sato K, Horton P, Frith MC. Adaptive seeds tame genomic sequence comparison. Genome Res. 2011;21(3):487–93.
23. Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL. Alignment of whole genomes. Nucleic Acids Res. 1999;27(11):2369–76.
24. Brudno M, Chapman M, Gottgens B, Batzoglou S, Morgenstern B. Fast and sensitive multiple alignment of large genomic sequences. BMC Bioinformatics. 2003;4:66.
25. Zhang P, Wei Y, Tian Q, Zou Q, Wang Y. Fast sequence alignment for centromere with RaMA. Genome Res. 2025;35(5):1209–18.
26. Bzikadze AV, Pevzner PA. UniAligner: a parameter-free framework for fast sequence alignment. Nat Methods. 2023;20(9):1346–54.
27. Kent WJ, Baertsch R, Hinrichs A, Miller W, Haussler D. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. Proc Natl Acad Sci USA. 2003;100(20):11484–9.
28. Treangen TJ, Messeguer X. M-GCAT: interactively and efficiently constructing large-scale multiple genome comparison frameworks in closely related species. BMC Bioinformatics. 2006;7:433.
29. Darling AE, Mau B, Perna NT. ProgressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. PLoS ONE. 2010;5(6):e11147.
30. Angiuoli SV, Salzberg SL. Mugsy: fast multiple alignment of closely related whole genomes. Bioinformatics. 2011;27(3):334–42.
31. Armstrong J, Hickey G, Diekhans M, Fiddes IT, Novak AM, Deran A, et al. Progressive cactus is a multiple-genome aligner for the thousand-genome era. Nature. 2020;587(7833):246–51.
32. Naznooshsadat E, Elham P, Ali SZ. FAME: fast and memory efficient multiple sequences alignment tool through compatible chain of roots. Bioinformatics. 2020;36(12):3662–8.
33. Treangen TJ, Ondov BD, Koren S, Phillippy AM. The harvest suite for rapid core-genome alignment and visualization of thousands of intraspecific microbial genomes. Genome Biol. 2014;15(11):524.
34. Bray N, Pachter L. MAVID: constrained ancestral alignment of multiple sequences. Genome Res. 2004;14(4):693–9.
35. Brudno M, Do CB, Cooper GM, Kim MF, Davydov E, Program NCS, et al. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. Genome Res. 2003;13(4):721–31.
36. Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol. 2000;302(1):205–17.
37. Paten B, Herrero J, Beal K, Fitzgerald S, Birney E. Enredo and pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. Genome Res. 2008;18(11):1814–28.
38. Morgenstern B, Frech K, Dress A, Werner T. Dialign: finding local similarities by multiple sequence alignment. Bioinformatics. 1998;14(3):290–4.
39. Bradley RK, Roberts A, Smoot M, Juvekar S, Do J, Dewey C, et al. Fast statistical alignment. PLoS Comput Biol. 2009;5(5):e1000392.
40. Altschul SF. Gap costs for multiple sequence alignment. J Theor Biol. 1989;138(3):297–309.
41. Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32(5):1792–7.
42. Earl D, Nguyen N, Hickey G, Harris RS, Fitzgerald S, Beal K, et al. Alignathon: a competitive assessment of whole-genome alignment methods. Genome Res. 2014;24(12):2077–89.
43. Fletcher W, Yang Z. Indelible: a flexible simulator of biological sequence evolution. Mol Biol Evol. 2009;26(8):1879–88.
44. Tang F, Chao J, Wei Y, Yang F, Zhai Y, Xu L, et al. HAlign 3: fast multiple alignment of ultra-large numbers of similar DNA/RNA sequences. Mol Biol Evol. 2022;39(8):msac166.
45. Katoh K, Standley DM. Mafft multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol. 2013;30(4):772–80.
46. Sievers F, Higgins DG. Clustal Omega, accurate alignment of very large numbers of sequences. Multiple Seq Alignment Methods. 2014;1079:105–16.
47. Kille B, Nute MG, Huang V, Kim E, Phillippy AM, Treangen TJ. Parsnp 2.0: scalable core-genome alignment for massive microbial datasets. Bioinformatics. 2024;40(5):btae311.
48. Lassmann T. Kalign 3: multiple sequence alignment of large datasets. Oxford University Press; 2020.
49. Ahrenfeldt J, Skaarup C, Hasman H, Pedersen AG, Aarestrup FM, Lund O. Bacterial whole genome-based phylogeny: construction of a new benchmarking dataset and assessment of some existing methods. BMC Genomics. 2017;18(1):19.

50. Zhang C, Nielsen R, Mirarab S. ASTER: A package for Large-Scale phylogenomic reconstructions. Mol Biol Evol. 2025;42(8):msaf172.
51. Zhang C, Nielsen R, Mirarab S. CASTER: direct species tree inference from whole-genome alignments. Science. 2025;387(6737):eadk9688.
52. Kuderna LF, Gao H, Janiak MC, Kuhlwilm M, Orkin JD, Bataillon T, et al. A global catalog of whole-genome diversity from 233 primate species. Science. 2023;380(6648):906–13.
53. Zhang C, Nielsen R. WASTER: Practical de novo phylogenomics from low-coverage short reads. bioRxiv [Preprint]. 2025:2025.01.20.633983. https://doi.org/10.1101/2025.01.20.633983.
54. Wei Y, Zou Q, Tang F, Yu L. WMSA: a novel method for multiple sequence alignment of DNA sequences. Bioinformatics. 2022;38(22):5019–25.
55. Mori Y. Libdivsufsort, a software library that implements a lightweight suffix array construction algorithm. 2015. Github; https://github.com/y-256/libdivsufsort.
56. Appleby A. SMHasher: a test suite designed to test the distribution, collision, and performance properties of non-cryptographic hash functions. 2016.
57. Bloom BH. Space/time trade-offs in hash coding with allowable errors. Commun ACM. 1970;13(7):422–6.
58. Zhang P, Zhou T, Zou Q, Luo X, HAlign-G. A rapid and low-memory multiple-genome aligner for large-scale closely related genomes. 2025. Github; https://github.com/malabz/HAlign-G.
59. Zhang P, Zhou T, Zou Q, Luo X. Dataset for HAlign-G: A rapid and low-memory multiple-genome aligner for large-scale closely related genomes. 2025. Zenodo; https://zenodo.org/records/17042774


## Publisher's Note