Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Sequence Analysis

Sequence Analysis

wfmash: whole-chromosome pairwise alignment using the hierarchical wavefront algorithm

Andrea Guarracino [1], Njagi Mwaniki [2], Santiago Marco-Sola [3, 4], and Erik Garrison [5]*

[1]Genomics Research Centre, Human Technopole, Milan, Italy
[2]???, ???, Pisa, Italy
[3]Department of Computer Sciences, Barcelona Supercomputing Center, Barcelona 08034, Spain
[4]Departament d'Arquitectura de Computadors i Sistemes Operatius, Universitat Autònoma de Barcelona, Barcelona 08193, Spain
[5]University of Tennessee Health Science Center, Memphis, TN, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Pairwise alignment of sequences is an important step in many bioinformatics analyses, including pangenome construction. Pangenomes are sequence models able to provide a full representation of the mutual alignment of collections of genomes. The time and memory required to compute pairwise alignments increase quadratically with the sequence length, making it impractical to directly align very long sequences without applying heuristic approaches to first determine possible syntenic regions in the sequences. Nevertheless, with the advances in sequencing technologies, new genome assemblies are produced at a high rate, pressing for the development of tools able to align sequences of the order of tens of megabases long.

Results: Here we present wfmash, a new gap-affine pairwise aligner designed to align DNA sequences at a whole-chromosome scale. wfmash applies a hierarchical implementation of the wavefront alignment algorithm to guide the alignment of very long sequences. It scales efficiently to large eukaryotic chromosomes, allowing users to perform pairwise alignment of thousands of large genomes using little time and memory.

Availability: wfmash is published as free software under the MIT open source license. Source code and documentation are available at `https://github.com/ekg/wfmash`. wfmash can be installed via Bioconda `https://bioconda.github.io/recipes/wfmash/README.html` or GNU Guix `https://github.com/ekg/guix-genomics/blob/master/wfmash.scm`.

Contact: egarris5@uthsc.edu

1 Introduction

Many biological analyses take advantage of aligning DNA sequences, ranging from read mapping (**???**) to variant detection (**?**), as well as de novo assembly (**?**) and pangenome construction (**??**). Pairwise sequence alignment can be applied to identify similar regions that may indicate functional, structural, and/or evolutionary relationships between two biological sequences. In this context, pangenomes model the full set of genomic elements in a given species or clade (**?**). These data structures encode the mutual relationships between all the genomes represented, in contrast to reference-based approaches which relate sequences to a particular genome chosen as reference. The unbiased approach allows studying the entire genetic diversity of a population. This opportunity, in combination with the massive amount of data available nowadays, presses for the development of tools able to compute pairwise alignment of very long sequences.

Popular seed-and-extend mapping methods, like Minimap2 (**?**), poorly scale in runtime and memory when generating sensitive alignments for chromosome-scale contigs available thanks to the new sequencing technologies (**?**). To scale to vertebrate genomes, methods for pangenome construction based on these approaches must first filter out centromeric and other highly-repetitive sequences, in contrast to the pangenome idea of modeling the full genetic variation in the samples.

In the interest of operating on the full pangenome, and to face the upcoming challenges in pangenome construction, we have developed wfmash, a new gap-affine pairwise tool for aligning DNA sequences at

a whole-chromosome scale. It supports split-read alignment and gap-affine penalties for insertion and deletions. wfmash scales efficiently to large eukaryotic chromosomes while requiring little time and memory, making it suitable to be applied on the pairwise alignment of hundreds of genomes. wfmash is a DNA sequence aligner based on the integration of MashMap (**?**), a fast approximate sequence mapper for computing local alignment boundaries between long DNA sequences, and the wavefront alignment algorithm (WFA) (**?**), an exact gap-affine algorithm that takes advantage of homologous regions between the sequences to accelerate the alignment process.

2 Methods

wfmash first applies a locality-sensitive hashing, from MashMap, to rapidly determine syntenic region boundaries between long DNA sequences. Then, a hierarchical implementation of the WFA allows computing the base-level global alignment of the identified mappings.

2.1 Approximate mapping

Each query sequence is broken into non-overlapping pieces of the requested length. These segments are then mapped using MashMap's sliding MinHash mapping algorithm (**?**). We extended MashMap, incorporating the robust winnowing (**?**) in the minimizer downsampling. Robust sampling avoids taking too many minimizers in low-complexity substrings, yielding improvements in runtime and memory-usage without affecting accuracy (**?**). Futhermore, we made it possible to set the number of mappings to return for each segment; this is useful for identifying paralogous regions and homologous relationship between sets of genomes. Together with the segment length and the minimum segment estimated identity, these settings allow users to precisely define the mapping space to consider, specifying the characteristics of homologies to compute.
FIXME THINGS ON THE FILTERING???
FIXME THING ON THE SPLIT READ AND/OR THE MERGING???

Finally, each mapping location is used as a target for base-level alignment using a hierarchical implementation of the WFA.

2.2 Base-level alignment

The time and memory required to compute the base-level alignment increases quadratically with the sequence length. The WFA provides an efficient way to decrease the amount of computation required to obtain the optimal alignment between two sequences, reducing the cost to be quadratic in the alignment penalty score of the optimal alignment (**?**). This means that the algorithm is very efficient in aligning similar homologous sequences (i.e., sequences with a low alignment penalty score), but high divergence genomes and/or noisy long reads can increase its memory and runtime costs.

To avoid such limitation, wfmash applies a hierarchical WFA, exploiting the WFA to guide the alignment process, but keeping the largest alignment problem size small. Rather than directly aligning the whole sequences, it aligns them to each other in small pieces W-bps long by applying a global WFA alignment (Figure 1). The whole global alignment is then computed over the full dynamic-programming matrix (high order DP-matrix) at W-bps resolution. Each cell in the high order DP-matrix corresponds to the alignment of a specific pair of fragments W-bps long from the two sequences to align. To determine if each cell is a match, and then guiding the W-bps resolution alignment, the global alignment using the standard WFA is performed between the two fragments. To accelerate the process, an alignment-free comparison of the two fragments is performed first by computing the mash distance between them. Cells whose fragments have a high mash distance are considered as mismatches, avoiding computing the corresponding base-level alignments.

Finally, the W-bps resolution traceback is applied to determine the set of base-level alignments on the optimal path in the high order DP-matrix. During the process, a further step is performed for refining the alignment. Inaccurate mapping estimation can lead to under-alignment at the beginning and the end of the aligned syntenic region. Therefore, WFA is applied by computing semi-global alignments to resolve the mapping boundaries. Moreover, the standard WFA is applied to resolve breakpoints of the structural variants when present in the aligned region.

The hierarchical implementation requires only the memory to align the sequences at W-bps resolution, limiting the runtime and the memory of the standard WFA by applying it to fragments W-bps long. This approach is more FIXME flexible than using a fixed-width band (it effectively has a variable bandwidth), requires no heuristic seeding step, and can benefit from parallel exploration of the wavefront.

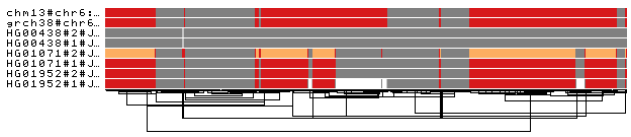

Fig. 1: PLACEHOLDER IMAGE XXX XXX XXX

3 Results

We evaluated the performance of wfmash `90ea671` against Minimap2-v2.20-r1061 (**?**) and Winnomap 2.03 (**?**), in terms of runtime and memory usage. The comparison was performed on both simulated and real datasets. Simulated data allowed evaluating the alignment quality by calculating precision and recall in variant identification. The evaluations were performed on a workstation running Ubuntu 20.04 equipped with 64 GB of RAM and an Intel i9-9900K CPU with 8 cores (16 threads). All commands are provided in the Supplementary Material.

3.1 Simulated data

We simulated long sequences from the *Saccharomyces cerevisiae* chromosome IV (S288C strain) and the telomere-to-telomere (T2T) assemblies of the human chromosomes 8, at different levels of length and divergence ([FIXME: MAYBETable 1 and Table 2]). Chromosomes were split into shorter sequences using the splitfa tool (6a0bb67) (**?**). In each run, all sequences have the same length, with 50% of them in reverse complement respect to the source chromosome. Single nucleotide variants (SNVs), small indels, and structural variants (SVs) were introduced using the Mutation-Simulator script (**?**).

To evaluate the correctness of the alignments, in each run we used the pairwise alignment of the input sequences to build a pangenome graph and identify the variants embedded in it. The graphs were built using (**?**). seqwish implements a lossless conversion from pairwise alignments between sequences to a pangenome graph encoding the sequences and their alignments. We called SNVs, small indels, and SVs using vg deconstruct (**?**), evaluating the false-negative and false-positive rates using vcfeval (**?**) for SNVs and small indels, and FIXME XXXXX for SVs.

3.2 Real data

XXXXX

4 Discussion

We implemented a novel gap-affine pairwise aligner, wfmash, to accelerate the computation of the mutual alignment of collections of genomes, a step required for constructing pangenome models. We have demonstrated that it efficiently performs with contigs representing full human chromosomes of 88 phased haplotypes from the Human Pangenome Reference Consortium year 1 assembly. Indeed, thanks to its efficiency, wfmash is already successfully applied in our pangenome graphs building pipeline (Garrison *et al.*, 2021). No less important, pairwise alignment is a central step of many bioinformatics applications, making our aligner a scalable solution to face the increasing yields of sequencing technologies in the coming years.

Acknowledgements

We thank members of the HPRC Pangenome Working Group for their insightful discussion and feedback, and members of the HPRC production teams for their development of resources used in our exposition.

Funding

We gratefully acknowledge support from NIH/NIDA U01DA047638 and NIH/NIGMS R01GM123489 (EG).

Data availability

Code and links to data resources used to build this manuscript and its figures, can be found in the paper's public repository: `https://github.com/AndreaGuarracino/wfmash-paper`.

References

Armstrong, J. *et al.* (2020). Progressive Cactus is a Multiple-Genome Aligner for the Thousand-Genome Era. *Nature*, **587**(7833), 246–251.

Baaijens, J. A. *et al.* (2019). Full-length de novo viral quasispecies assembly through variation graph construction. *Bioinformatics*, **35**(24), 5086–5094.

Beyer, W. *et al.* (2019). Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics*, **35**(24), 5318–5320.

Consortium, C. P. (2018). Computational pan-genomics: status, promises and challenges. *Briefings in Bioinformatics*, **19**(1), 118–135.

Durant, E. *et al.* (2021). Panache: a web browser-based viewer for linearized pangenomes. *Bioinformatics*.

Eizenga, J. M. *et al.* (2020a). Efficient dynamic variation graphs. *Bioinformatics*, **36**(21), 5139–5144.

Eizenga, J. M. *et al.* (2020b). Pangenome graphs. *Annual Review of Genomics and Human Genetics*, **21**(1), 139–162.

Ewels, P. *et al.* (2016). MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, **32**(19), 3047–3048.

Garrison, E. (2019). Graphical pangenomics.

Garrison, E. *et al.* (2018). Variation Graph Toolkit Improves Read Mapping by Representing Genetic Variation in the Reference. *Nature Biotechnology*, **36**(9), 875–879.

Garrison, E. *et al.* (2021). The pangenome graph builder. `https://github.com/pangenome/pggb`.

GFA Working Group (2016). Graphical fragment assembly (gfa) format specification. `https://github.com/GFA-spec/GFA-spec`.

Gonnella, G. *et al.* (2018). GfaViz: flexible and interactive visualization of GFA sequence graphs. *Bioinformatics*, **35**(16), 2853–2855.

Grasso, C. and Lee, C. (2004). Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics*, **20**(10), 1546–1556.

Guarracino, A. *et al.* (2021). wfmash: whole-chromosome pairwise alignment using the hierarchical wavefront algorithm. `https://github.com/ekg/wfmash`.

Hickey, G. *et al.* (2020). Genotyping Structural Variants in Pangenome Graphs Using the vg Toolkit. *Genome Biology*, **21**(1), 35.

Kehr, B. *et al.* (2014). Genome alignment with graph data structures: a comparison. *BMC Bioinformatics*, **15**(1).

Lee, C. *et al.* (2002). Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**(3), 452–464.

Li, H. *et al.* (2009). The sequence alignment/map format and samtools. *Bioinformatics*, **25**(16), 2078–2079. 19505943[pmid].

Li, H. *et al.* (2020). The Design and Construction of Reference Pangenome Graphs with Minigraph. *Genome Biology*, **21**(1), 265.

Liang, Q. and Lonardi, S. (2021). Reference-agnostic representation and visualization of pan-genomes. *BMC Bioinformatics*, **22**(1), 502.

Logsdon, G. A. *et al.* (2021). The structure, function and evolution of a complete human chromosome 8. *Nature*, **593**(7857), 101–107.

Miga, K. H. *et al.* (2020). Telomere-to-Telomere Assembly of a Complete Human X Chromosome. *Nature*, **585**(7823), 79–84.

Nance, M. A. *et al.* (1999). Analysis of a very large trinucleotide repeat in a patient with juvenile Huntington's disease. *Neurology*, **52**(2), 392–394.

Neueder, A. *et al.* (2017). The pathogenic exon 1 HTT protein is produced by incomplete splicing in Huntington's disease patients. *Scientific Reports*, **7**(1), 1307.

Niu, F. *et al.* (2011). Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *arXiv preprint arXiv:1106.5730*.

Nurk, S. *et al.* (2021). The complete sequence of a human genome. *BioRxiv*.

Paten, B. *et al.* (2017). Genome Graphs and the Evolution of Genome Inference. *Genome Research*, **27**(5), 665–676.

Pevzner, P. A. (2004). De novo repeat classification and fragment assembly. *Genome Research*, **14**(9), 1786–1796.

Piovesan, A. *et al.* (2019). On the length, weight and gc content of the human genome. *BMC Research Notes*, **12**(1), 106.

Prezza, N. (2017). A framework of dynamic data structures for string processing. *arXiv preprint arXiv:1701.07238*.

Quinlan, A. R. and Hall, I. M. (2010). Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**(6), 841–842.

Sekar, A. *et al.* (2016). Schizophrenia risk from complex variation of complement component 4. *Nature*, **530**(7589), 177–183.

Sibbesen, J. A. *et al.* (2021). Haplotype-aware pantranscriptome analyses using spliced pangenome graphs. *BioRxiv*.

Siren, J. *et al.* (2020). Haplotype-aware graph indexes. *Bioinformatics*, **36**(2), 400–407.

Wick, R. R. *et al.* (2015). Bandage: interactive visualization ofde novogenome assemblies: Fig. 1. *Bioinformatics*, **31**(20), 3350–3352.

Yokoyama, T. T. *et al.* (2019). MoMI-G: modular multi-scale integrated genome graph browser. *BMC Bioinformatics*, **20**(1), 548.

Zheng, J. X. *et al.* (2018). Graph drawing by stochastic gradient descent. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, **25**(9), 2738–2748.