



Francesco Andreace

***k*-mers to rule them all**

Analysis of human pangenome graphs and
other *k*-mer-based applications

Thesis submitted for the degree of Philosophiae Doctor

Department of Computational Biology
Institut Pasteur Paris, Université de Paris Cité

Edited doctoral school, Sorbonne Université

2024



© Francesco Andreace, 2024

*Series of dissertations submitted to the
Institut Pasteur Paris, Universite' de Paris Cite, Sorbonne Universite'
No. 1234*

ISSN 1234-5678

All rights reserved. No part of this publication may be
reproduced or transmitted, in any form or by any means, without permission.

Cover: TBD - in PHDUIO.cls.
Print production: Pasteur Paris.

To Sofia

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at Sorbonne Université. The research presented here was conducted at the Institut Pasteur, under the supervision of Doc. Rayan Chikhi and Doc. Yoann Dufresne. This work was supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 956 229 (+Pasteur + INCEPTION).

The thesis is a collection of the published paper I am first author of, novel elaboration of the work I did for other papers where I am not first author and presentation of work that has or will be submitted to revision and is then not yet published. The common theme to them is the development k -mer-based methods and the use of such for (meta/pan/-)genomics. The paper is preceded by an introductory chapter that provides background information and motivation for the work and is succeeded by considerations and perspectives.

Acknowledgements

Thanks for your consideration.

Francesco Andreace

Paris, September 2024

Contents

Preface	iii
Contents	v
List of Figures	vii
List of Tables	ix
1 Pushing the limit of pan-genome construction methods	1
1.1 Motivation	1
2 Comparing methods for constructing and representing human pangenome graphs	3
2.1 Introduction	4
2.2 Results	5
2.3 Discussion	16
2.4 Conclusions	17
2.5 Methods	18
2.6 Perspectives	22
2.7 Building a <i>Lodderomyces elongisporus</i> pangenome reference: overcoming current limitations.	24
2.8 Conclusion and Perspectives	29
Bibliography	33
Papers	40
Appendices	41

List of Figures

2.1	The complete pangenome construction scheme and visualization. A , The overall workflow, using 5 different tools on 3 different datasets; B , complete 104 haplotypes variation graph built by Minigraph ; C , focus on part of HLA (MHC) region in chromosome 6 from panel B; D , focus on DRB1-5 locus of HLA from panel C; E , complete 10 haplotypes variation graph built with pggb ; F , 10 haplotypes variation graph built with Minigraph-Cactus ; G , 104 haplotypes pangenome mdbg ; H , 10 haplotypes Bifrost dBG. All graphs except those produced by Minigraph have been simplified using gfatools and rendered using Bandage . VG is for variation graph.	7
2.2	Representations of the HLA-E locus by five graph construction methods over three increasing large human pangenomes. Nodes highlighted in red contain part of the locus sequence. The numbers of nodes and edges displayed below each graph concerns the whole subgraph (both red and grey nodes). Minigraph , on H2, H10 and H104, and mdbg , on H2, have only a portion of one node highlighted since the 4.8k bp region is contained inside a single, long node.	11
2.3	Representations of the complex HLA region by five graph construction methods over three increasing large human pangenomes. See caption of Fig. 2.2 for details.	13
2.4	cDBG representation and phylogeny analysis of the <i>Lodderomyces elongisporus</i> pangenome.	26
2.5	Sequence Identity of contigs assigned to reference chromosomes. Image produced using a pipeline developed by Simon Heumos.	27
2.6	Gfaestus visualization of the pggb variation graph of the 11 <i>Lodderomyces elongisporus</i> samples. Image produced by Simon Heumos.	28
2.7	Difference in output between Minigraph and Minigraph-Cactus of the chimeric graph produced to visualize the inter-chromosomal event between C,G and H.	29
2.8	Community partition of the contigs based on alignment scores.	30
2.9	plotsr visualization of the inter-chromosomal recombination detected using wfmash and SyRI	30

List of Figures

2.10	The tangle of chromosomes C, G and H in the Minigraph-Cactus variation graph. Nodes are colored based on Minigraph alignment of chromosome C (dark green), G (blue) and H (red) of the reference assembly B2. The three chromosomes are bound together because of the construction.	31
2.11	1D visualization of differences between pggb and Minigraph-Cactus output.	31

List of Tables

2.1	Time, memory, final disk space, nodes, edges, total connected components and connected components with more than 1% of base pairs comparison of Bifrost , mdbg , pggb , Minigraph and Minigraph-Cactus for different number of haplotypes in input. Minigraph-Cactus times include the Minigraph graph construction step. pggb was not able to complete its execution on the largest dataset in more than 2 weeks thus it is not considered. Minigraph-Cactus failed to compute the 104 HAP dataset. . .	9
2.2	Relative strengths of five pangenome graph construction tools Explanation of rows: 1) efficacy of construction algorithm, measuring wall-clock time; 2) degree to which variants (e.g. SNPs) are retained; 3) ability of a tool to perform well on large datasets, both in comparison to other tools but also compared to smaller datasets; 4) ability to modify the produced data structure to add or remove haplotypes; 5) property of producing the same result irrespective of perturbations, such as permutation of the input order, and repeating the same run; 6) existence of tools (and operations) that can be applied to the resulting graphs; 7) whether input haplotypes information is retained by the tools, and if so, its space efficiency; 8) whether the entire graph can be directly visualized and interpreted; 9) easiness of 'zooming in' a specific genomic region and interpret variants; 10) summarizes the functionalities provided by the tools to annotate the pangenomes with genomic data; 11) ability to share information between the graph and a linear reference.	16
2.3	Description of the three datasets generated to test the scalability of the tools Data sources: ¹ Google Brain Genomics [27]; ² Human Pangenome Reference Consortium [28]; ³ 1000 Genomes Project [28]; ⁴ Telomere to Telomere Consortium [28].	19
2.4	URL, version, pangenome representation and parameters of the three analyzed tools. pggb/0.2.0 used wfsmash v0.7.0, seqwish v0.7.3 and smoothxg v0.6.1.	20

Chapter 1

Pushing the limit of pan-genome construction methods

In this first chapter we present and discuss two situations in which we have pushed the limit of pan-genome constructing methods. In the first one we analyzed the current state of the art methods available at the moment and stress-tested them to generate what was, to the best of our knowledge, the largest human pan-genome produced at the time. The second one was the generation of a yeast pan-genome reference for the species *Lodderomyces elongisporus* to investigate cross-chromosomal rearrangements. In order to best capture the event between 3 chromosomes, we had to modify one of the most used pangenome construction pipeline. Here we present the challenges faced and discuss about how to achieve balance between biological correctness, genome variation resolution, scalability and downstream applications of the pan-genome.

1.1 Motivation

The following paper originates from a discussion early in my PhD journey, on which are the best suited tools for large cohort pangenomes of species with large genomes, like animal or pants. As pointed out in the introduction, there is no one-fits-all solution and most of the tools, at the time of the analysis, were freshly released or distributed under development. It was therefore important for the community of developers and users of such new tools and models to understand the limitations and the potential of the new pangenomics methods. In order to evaluate the current state of the art of pangenome building tools, we decided to perform a thorough assessment of the best available methods by giving as input the largest dataset we could produce to mimic the conditions that they could be required to be used in the near future. We decided to test on human data because of the importance of understanding how pangenomes can be used and adapted to current genomic studies.

As introduced in section ??, there are multiple ways of representing a group of genomes to be analyzed or used jointly. One that took traction in the last few years has been graphs. Graphs can represent the sequences as labels of nodes, relationship between them (adjacency or overlap) as edges and infer difference in the genomes as different set of nodes in them.

We specifically focused our attention on the most used graph models: variation graphs and De Bruijn Graphs. In variation graph edges represent adjacencies, i.e the genome is spelled by a walk on nodes connected by an edge. In De Bruijn Graphs they represent overlaps, i.e. the suffix of a node is the prefix of the next node connected to it: this implies that edges can exists between nodes that are

1. Pushing the limit of pan-genome construction methods

not adjacent in the genome. As discussed in the next sessions, this distinction implies several differences in how these graphs can be used for downstream analysis.

In this article, we surveyed the the methods and tools that build such graphs, then tested them on different dataset sizes and permutations, and finally analyzed the resulting representation's features. The outcome is a small guide on which are the best applications for each of these tools and an analysis of how they represent variations in genomes.

Chapter 2

Comparing methods for constructing and representing human pangenome graphs

Francesco Andreace, Pierre Lechat, Yoann Dufresne, Rayan Chikhi

Published in *Genome Biology*, November 2023, volume 24, issue 1, article number 274. DOI: 10.1186/s13059-023-03098-2.

Abstract

Background: As a single reference genome cannot possibly represent all the variation present across human individuals, pangenome graphs have been introduced to incorporate population diversity within a wide range of genomic analyses. Several data structures have been proposed for representing collections of genomes as pangenomes, in particular graphs.

Results: In this work we collect all publicly available high-quality human haplotypes and construct the largest human pangenome graphs to date, incorporating 52 individuals in addition to two synthetic references (CHM13 and GRCh38). We build variation graphs and de Bruijn graphs of this collection using five of the state-of-the-art tools: **Bifrost**, **mdbg**, **Minigraph**, **Minigraph-Cactus** and **pggb**. We examine differences in the way each of these tools represents variations between input sequences, both in terms of overall graph structure and representation of specific genetic loci.

Conclusion: This work sheds light on key differences between pangenome graph representations, informing end-users on how to select the most appropriate graph type for their application.

Contents

2.1	Introduction	4
2.2	Results	5
2.3	Discussion	16
2.4	Conclusions	17
2.5	Methods	18
2.6	Perspectives	22

2. Comparing methods for constructing and representing human pangenome graphs

2.7	Building a <i>Lodderomyces elongisporus</i> pangenome reference: overcoming current limitations.	24
2.8	Conclusion and Perspectives	29

2.1 Introduction

In recent years, the majority of studies on human genetics have been conducted on the basis of comparing new samples against a single, standard reference sequence. This reference sequence is a linear succession of nucleotides that acts as a blueprint of the human genome. It is routinely used to align raw sequencing data to it in order to find variations between genomes, e.g. single-nucleotide polymorphisms (SNPs), insertions or deletions (indels). It also is the backbone of the UCSC Genome Browser [1] which enables inspection of genomic and epigenomic features. Despite updates that have improved the quality of the human reference sequence in the last two decades, its linear form severely limits the ability to capture population genetic diversity. For instance the locations of frequently observed structural variations cannot be easily integrated into a linear reference. To see this, consider the difficulty of designing a suitable coordinate system in the presence of (possibly nested) structural variants. Having a single genome as reference sequence also introduces an observational bias towards the chosen alleles that were integrated into that sequence, negatively impacting many primary analyses such as reads mapping, variant calling, genotyping and haplotype phasing. As a result our ability to precisely characterize structural variants, SNPs and small indels is limited [2, 3, 4]. The GRCh38 human reference genome is estimated to miss up to 10% of our species genetic information [5].

Improvements in sequencing data quality and length, as well as genome assembly methods, are providing a fast expanding collection of haplotype-resolved human genome assemblies. If adequately combined together, these high-quality individual genomes may offer a powerful alternative to the linear reference. There now is an active line of research on pangenomes, i.e. data structures that represent a collection of genomic sequences to be analyzed jointly or to form a reference [3, 6].

Pangenome-based approaches have been shown to improve biological analyses. Pangenomes are at the basis of bioinformatics tools that perform high-quality short read mapping [4], genotyping of SNPs, indels and SVs [7], RNA-seq mapping [8]; de novo variant calling [2]; to store, compress and retrieve high quality genomes [9]; to condensate all the information from a high number of genomes to then visualize specific regions or perform ad-hoc analysis, particularly on complex loci, SVs and tandem repeats [8]. These results pave the way for new applications, e.g. genome-wide association studies, where more precise identification of variants can improve the scope of genetic studies in aging, human diseases, and cancer [3, 6].

Several pangenomic data structures have been proposed: multiple sequence alignments, de Bruijn graphs, cyclic and acyclic variation graphs, and haplotype-centric models that use the Burrows-Wheeler transform [3]. Each of these

approaches aim to represent a collection of genomic sequences in an efficient way, to store, visualize, and retrieve differences of interest between the considered genomes. Graph-based pangenome data structures, such as the de Bruijn graph and the variation graph, appear so far to be the most advanced in their ability to handle large amounts of input data. They are capable of representing tens to hundreds of human haplotypes simultaneously. Variations graphs use a sequence graph and a list of paths to store input haplotypes, while de Bruijn graphs store all haplotype k -mers annotated by their haplotype(s) of origin. Scaling pangenome graph data structures to store hundreds of genomes is a challenge that requires significant computational resources and engineering efforts. Many software tools have been created, here we briefly describe major ones. Pantools [10] and Bifrost [11] are two methods that have been developed to generate pangenomes for analysis on large collections of genomes, mostly for applications in phylogenetics and bacterial genomics. The PanGenome Graph Builder (**pggb**) [12], **Minigraph-Cactus** and **TwoPaCo** [13] are methods for building general-purpose pangenome graphs. **Minigraph** [14] builds a particular type of pangenome graph by aligning sequences in an iterative way to a reference template. Minimizer-space de Bruijn graphs (**mdbg**) [15] are variants of de Bruijn graphs that can efficiently represent very large collections of bacterial pangenomes (e.g. 600,000 bacteria). **vg** [2] builds variation graphs from a reference sequence and a variant calling file (vcf) that contains a list of variations from it. Many human pangenomes have been generated, e.g. using Pantools [10] (7 genomes), **Minigraph** [14] (94 haplotypes), **Minigraph-Cactus** [16, 17] and **pggb** [8] (94 single chromosomes), and **TwoPaCo** [13] (100 simulated genomes). Lastly, a draft version of a human reference pangenome constructed using **pggb** and the **Minigraph-Cactus** pipeline has appeared in a very recent article from the Human Pangenome Reference Consortium [8]. These pangenomes are still limited by some factors: at the present moment, the number of high-quality haplotype assemblies is still low, even if it is expected to grow in the future; the vcf files containing variation are limited in term of bias, type of variation or number of samples; the population representation, even if opened up in recent years to more ethnicities, is still affected by sampling bias.

2.2 Results

In this article we provide a comprehensive view of whole-genome human pangenomics through the lens of five methods that each implement a different graph data structure: **Bifrost**, **mdbg**, **Minigraph**, **Minigraph-Cactus** and **pggb**. We examine several features of pangenome graphs, in particular their scalability and how they represent genetic diversity. To this end we collected all publicly available high-quality human haplotypes and attempted to construct pangenomes of various complexity with each selected tool. Although **vg** has been widely used at the basis of relevant pangenome-based discoveries, for example on fast and accurate short read mapping [4], we decided to not consider it in our analysis for two main reasons: the bias introduced by the reference sequence

2. Comparing methods for constructing and representing human pangenome graphs

that is used as the backbone of the graph (and associated to the vcf) together with the limited capacity of this method to integrate structural variations from many genomes. We believe both aspects are drivers of the use of pangenome graphs.

Scalability and characteristics of pangenome graph construction tools

We ran the above five tools on three datasets consisting of 2, 10, and 104 human haplotypes respectively (Table 2.3). We compared the computational performance of construction algorithms as well as characteristics of the produced pangenome graphs. The goal is to assess the ability of each method to scale to data available in the near future, i.e. thousands or even millions of human genomes [5].

The performance of each tool is evaluated in terms of running time, peak memory, disk space required by the output data structure (graph and annotations). We also compared the number of nodes, edges and connected components as indicators of the complexity of the graph. Results are displayed in Table 2.1.

In terms of running time, **mdbg** is two orders of magnitude faster than other tools on all considered datasets, taking around two minutes on the H2 dataset and half an hour on H104. **Bifrost** is the second fastest on H104 (18 hours), and **Minigraph** is the second fastest on H2 (8 minutes). **Minigraph-Cactus** takes one order of magnitude more time than **Minigraph**. We could not obtain graphs for **pggb** and **Minigraph-Cactus** on H104 as for the first execution did not finish after 2 weeks and the second returns an error.

In terms of memory usage, **mdbg** consistently uses less than half the memory of other tools (31 GB on H104), followed by **Minigraph** (61 GB on H104). On H2 all tools used between 8 and 66 GB of memory.

All tools used reasonable disk space to store the resulting graph, \leq 12 GB for H10 and \leq 38 GB for H104. Although **Minigraph-Cactus** and **pggb** retain all variations and are the only two tools able to reconstruct the input haplotypes directly from the graph, they are the second and third most efficient in term of disk space (for **Minigraph-Cactus**, 3.6 GB on H2 and 7 GB on H10). While **Bifrost** and **Minigraph** perform all computation in memory, **pggb**, **Minigraph-Cactus**, and **mdbg** store intermediate files on disk, taking comparable space to the input size (up to 3x for **Minigraph-Cactus**).

Different tools yield different pangenome graphs topologies

Graph metrics such as the number of nodes, edges and connected components provide useful insights on the level of detail of the represented variations and on the complexity and accessibility of the information inside the pangenome.

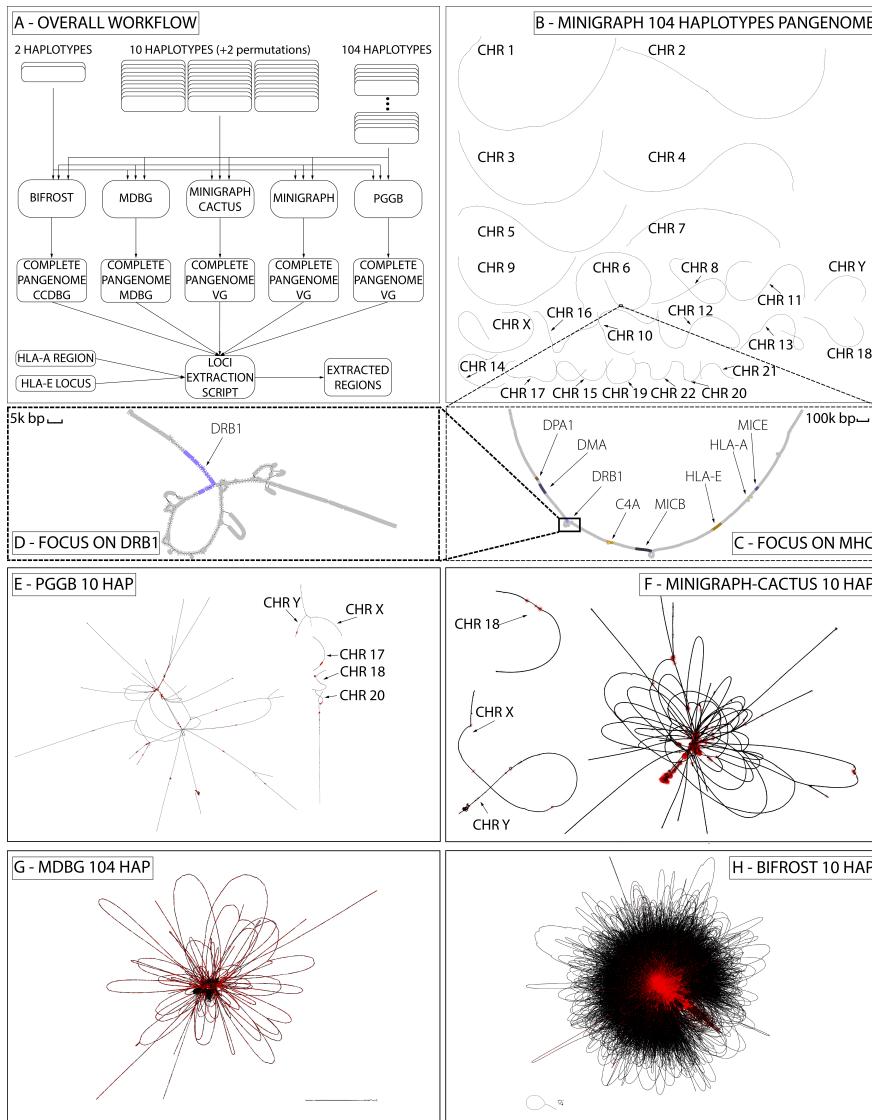


Figure 2.1: The complete pangenome construction scheme and visualization. **A**, The overall workflow, using 5 different tools on 3 different datasets; **B**, complete 104 haplotypes variation graph built by **Minigraph**; **C**, focus on part of HLA (MHC) region in chromosome 6 from panel B; **D**, focus on DRB1-5 locus of HLA from panel C; **E**, complete 10 haplotypes variation graph built with **pggb**; **F**, 10 haplotypes variation graph built with **Minigraph-Cactus**; **G**, 104 haplotypes pangenome **mdbg**; **H**, 10 haplotypes **Bifrost** dBG. All graphs except those produced by **Minigraph** have been simplified using gfatools and rendered using Bandage. VG is for variation graph.

2. Comparing methods for constructing and representing human pangenome graphs

The number of graph nodes varies between 17,000 and 11 millions for the H2 dataset across all tools. In all cases, the number of nodes is at least 3 orders of magnitude smaller than the number of bases in the haplotypes, indicating that pangenome graphs are effective at compressing linear parts of the haplotypes. Tools which discard variations (**Minigraph** and **mdbg**) yield in the order of 10^4 – 10^5 nodes across all datasets, while tools which retain all variation (**Bifrost**, **Minigraph-Cactus** and **pggb**) yield in the order of 10^6 – 10^7 nodes. In all cases going from the H10 dataset to the H104 dataset increases the number of nodes by 5x, indicating that graph complexity grows sublinearly with the number of added haplotypes.

The number of connected components varies between 2 and 1402 across all methods and datasets, and the number of large components (i.e. those with more than 1% of total base pairs) varies between 1 and 30. If chromosomes were separated perfectly, pangenome graphs should contain exactly 24 connected components (one per nuclear chromosome, excluding mitochondria). **Minigraph** produces 24 large connected components as the number of chromosomes in the reference CHM13 v2.0 (25 including mitochondria). **Bifrost** and **Minigraph-Cactus** yield graphs with less than 25 connected components while **mdbg** and **pggb** have more than 25. In the **Bifrost** dBG, the vast majority of sequences (>99.99%) are in a single giant component, as chromosomes are joined because they share common k -mers. In **mdbg** such joining does not occur on dataset H2, which has 24 large enough components (each containing > 1% of bases), possibly due to the absence of long and similar enough regions between chromosomes. **Minigraph** does not map any mitochondrial sequence from the input haplotypes to the reference, while they do get included into **Minigraph-Cactus** graphs.

Even if it is common practice to analyze pangomes chromosome by chromosome [8, 17], in this analysis we purposely used entire genomes as input instead. This was done for two reasons: i) to highlight the scalability of the tools, and ii) because separating chromosomes prevents the identification of inter-chromosomal inversions, translocations, and transposable elements, even if most of the generated inter-chromosomal events are probably alignment artifacts. The effects of this choice can be seen in the **pggb** and the **Minigraph-Cactus** H10 variation graphs of Figure 2.1. In the **pggb** graph 19 chromosomes are linked into a single giant component, while chromosomes 17, 18, 20, X, and Y are in other large components. This giant component consists of 25 M nodes that contain 83% of the total basepairs. The remaining 859 components represent only 4.7% of the total bases due to small sequences in the input haplotypes. In the **Minigraph-Cactus** graph all chromosomes are linked into a single giant component except chromosome 18 that is in a separate component, and the sexual chromosomes (X and Y) that are connected together into another component.

Table 2.1: Time, memory, final disk space, nodes, edges, total connected components and connected components with more than 1% of base pairs comparison of **Bifrost**, **mdbg**, **pggb**, **Minigraph** and **Minigraph-Cactus** for different number of haplotypes in input. **Minigraph-Cactus** times include the **Minigraph** graph construction step. **pggb** was not able to complete its execution on the largest dataset in more than 2 weeks thus it is not considered. **Minigraph-Cactus** failed to compute the 104 HAP dataset.

Haplotypes	Metric	Bifrost	pggb	Minigraph	Minigraph-Cactus	mdbg
2	time (hh:mm:ss)	1:21:25	15:45:30	00:08:33	3:11:59	00:02:38
	memory (GB)	53	24	38	66	8
	disk space (GB)	4.8	4.3	2.9	3.6	4.4
	nodes	9,482 k	8,492 k	34 k	10,851 k	17 k
	edges	13,108 k	11,503 k	48 k	14,702 k	23 k
	conn comp	2	1402	25	4	174
10	conn comp > 1% bp	1	30	24	4	24
	time (hh:mm:ss)	2:27:29	117:08:09	2:03:29	15:57:05	00:05:46
	memory (GB)	102	71	49	154	18
	disk space (GB)	12	7.6	2.9	7	9.7
	nodes	27,468 k	29,315 k	133 k	37,767 k	67 k
	edges	37,584 k	40,282 k	190 k	51,595 k	93 k
104	conn comp	3	864	25	3	40
	conn comp > 1% bp	1	5	24	3	1
	time (hh:mm:ss)	18:38:28	—	46:22:00	—	00:31:38
	memory (GB)	122	—	61	—	39
	disk space (GB)	29.4	—	3.2	—	38
	nodes	106,339 k	—	632 k	—	270 k
	edges	293,839 k	—	912 k	—	396 k
	conn comp	17	—	25	—	1097
	conn comp > 1% bp	1	—	24	—	1

Interpretation of variation in pangenome graphs: focus on two HLA loci

The ability to detect and annotate variations among input haplotypes defines the scope of each pangenome graph construction method. Previous work [18] recommends to build graphs on a specific loci rather than the entire genome for the purpose of i) identifying genomic diversity and ii) mapping raw reads to divergent regions, specifically difficult-to-map repeats. Here we evaluate how pangnomes built from entire haplotypes represent specific biologically relevant loci.

Extraction of HLA-E and a complex HLA region from complete pangenome graphs We extracted from complete pangnomes the regions corresponding to two loci of the Human Leukocyte Antigen complex, also known as HLA. These regions are highly medically relevant as they contain many disease-associated variants [19]. The first locus is the HLA-E gene, that is part of the nonclassical class I region genes, spanning 4,8 kbp and is relatively conserved across populations. It has been shown to have significant association with hospitalization and ICU admission as a result of COVID-19 infection [20]. The second is an HLA complex region comprising the HLA-A gene, part of the classical, highly

2. Comparing methods for constructing and representing human pangenome graphs

polymorphic class I region. It is around 58 kbp long and contains the HLA-U, HLA-K, HLA-H, and HCG4B genes. We extracted these two regions from pangenome graphs using a custom script that yields a subgraph corresponding to a given set of sequences and their variation. The script uses a different recommended method for each of the pangenome graph types. In a nutshell, we extracted regions using exact coordinates when possible, and resorted to sequence-to-graph alignment otherwise (see Appendix Section "Loci extraction method" for details). While on variation graphs and mDBGs nearby nodes of an aligned region correspond to variations of the locus, this is not always true for standard dBGs. Extracting accurate and complete loci representation is an unsolved challenge for dBGs.

HLA-E: a low complexity region Figure 2.2 shows how the different tools represent HLA-E over datasets H2, H10 and H104. As expected, **Minigraph** does not detect any variation, since the SNPs that characterize the region are too small to be considered in the construction steps of their algorithm. **pggb**, on the contrary, has 2 SNPs in H2 and 3 in H10. **Bifrost** detects the same SNPs as **pggb** in H2 and H10. Both of them represent the exact same variations and render the same haplotypes paths. **mdbg** captures the heterozygosity of a large region containing the HLA-E locus as the number of samples grows. As the **mdbg** graph is built in minimizer space, nodes represent long genomic segments (in the order of hundreds of thousand of base-pairs). In H10 and H104, the minimizer-space representations of the haplotypes are identical; however, differences in flanking regions of the graph create variations that are captured in extra nodes that are also extracted in this region. On H2, **Minigraph-Cactus** detects 3 variations as the dataset used is different, containing the CHM13 reference and just one haplotype of HG006 (as in **Minigraph**), as discussed in Section "Datasets and haplotypes collection".

Figure 2.2 also illustrates how pangenome complexity grows with the number of genomes: the **Bifrost** H104 subgraph has the most variation across all methods, highlighting that dBGs represent variations exhaustively in large graphs. On the other hand, **pggb** has the most straightforward method for extracting subgraphs, and also represents variants exhaustively in datasets H2 and H10, but could not scale to the H104 dataset.

HLA complex locus: high complexity region Figure 2.3 is the counterpart of Figure 2.2 for the complex locus part. In this case the overall interpretability of the region is more challenging, as the number and the structure of the variations is different than in HLA-E. It is also more difficult to compare across tools. Base-level variations, e.g. SNPs, are not visually recognizable in Figure 2.3 in the methods that retain them (i.e. **pggb**, **Minigraph-Cactus** and **Bifrost**) due to the large sizes of graphs.

There are notable differences in how tools represent the variation, which is well-illustrated in the H2 dataset. While **Minigraph** renders H2 as a single

HLA-E LOCUS COMPARISON

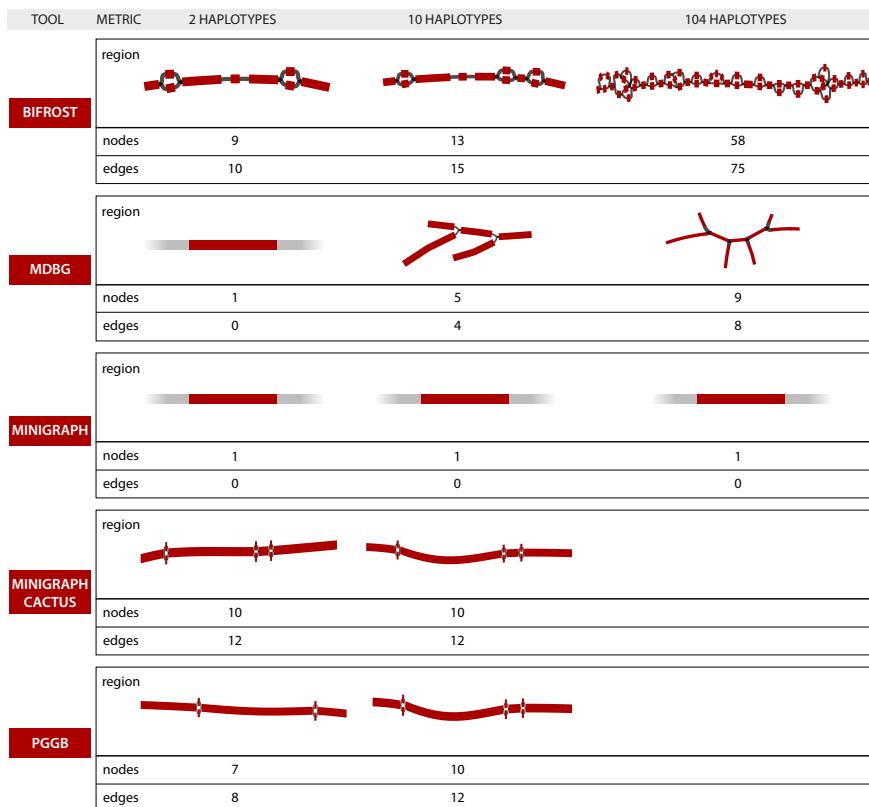


Figure 2.2: Representations of the HLA-E locus by five graph construction methods over three increasing large human pangenomes. Nodes highlighted in red contain part of the locus sequence. The numbers of nodes and edges displayed below each graph concerns the whole subgraph (both red and grey nodes). **Minigraph**, on H2, H10 and H104, and **mdbg**, on H2, have only a portion of one node highlighted since the 4.8k bp region is contained inside a single, long node.

2. Comparing methods for constructing and representing human pangenome graphs

sequence plus a large structural variant (SV) of $\approx 52\text{k bp}$, **pggb** separates it into two paths that differ by $\approx 54\text{k bp}$ in length. **Bifrost** represents a detailed bubble that contains many variations inside each path. In **mdbg**, even extracting the complete locus is a challenge as many of the subgraph nodes were not selected by our procedure. **Minigraph-Cactus** adds base level divergences between haplotypes on top of **Minigraph** SV graph.

These differences between representations are further accentuated in the H10 dataset. For it, **pggb** tends to separate the haplotypes into different paths, **Bifrost** renders consistently the same compacted representation and **Minigraph** neglects most of the small differences but is able to display accurately the general picture, and **Minigraph-Cactus**, as in H2, adds small variations on top of **Minigraph** structure.

Uncovering characteristics of graphical pangenome tools

The data structures generated by pangenome building tools are expected to facilitate comparisons between the input genomes. In addition pangenome graphs should be stored in such a way to be easily used by downstream applications. We identify 8 important features for pangenome graph construction tools: i) stability, ii) editability, iii) accessibility by downstream applications, iv) haplotype compression performance v) ease of visualization, vi) quality of metadata and annotation. Two other but important features, scalability and interpretability of produced graphs, were already discussed in Sections "Scalability and characteristics of pangenome graph construction tools" and "Interpretation of variation in pangenome graphs: focus on two HLA loci". Table 2.2 summarizes some of the following considerations on the relative strength of the tools.

Editability and dynamic updates As more high quality assemblies will be generated in the near future, haplotypes may be added to a pangenome, or replaced by improved versions. Updating an existing data structure instead of rebuilding it from scratch is both computationally and energetically efficient. However, many succinct data structures currently used in pangenome representation are static, i.e. cannot be updated. Some methods allow a restricted set of editing operations. **Minigraph** allows to add new haplotypes on top of an already built graph. **Bifrost** provides C++ APIs to add or remove (sub-)sequences, k -mers and colors from the ccdBG. **pggb**, using **odgi** [21], allows specific operations that delete and modify nodes and edges and add and modify paths through the graph. As **Minigraph-Cactus** can be opened with **odgi**, it supports the same operations as **pggb**. The current **mdbg** implementation uses a dynamic hash table, but does not expose an interface that supports updates.

Stability Counter-intuitively, a pangenome graph construction tool may in some cases generate different outputs when executed multiple times with the same haplotypes as input. This *unstability* could be due to a permutation in the order of the sequences given as input, or non-determinism in the construction

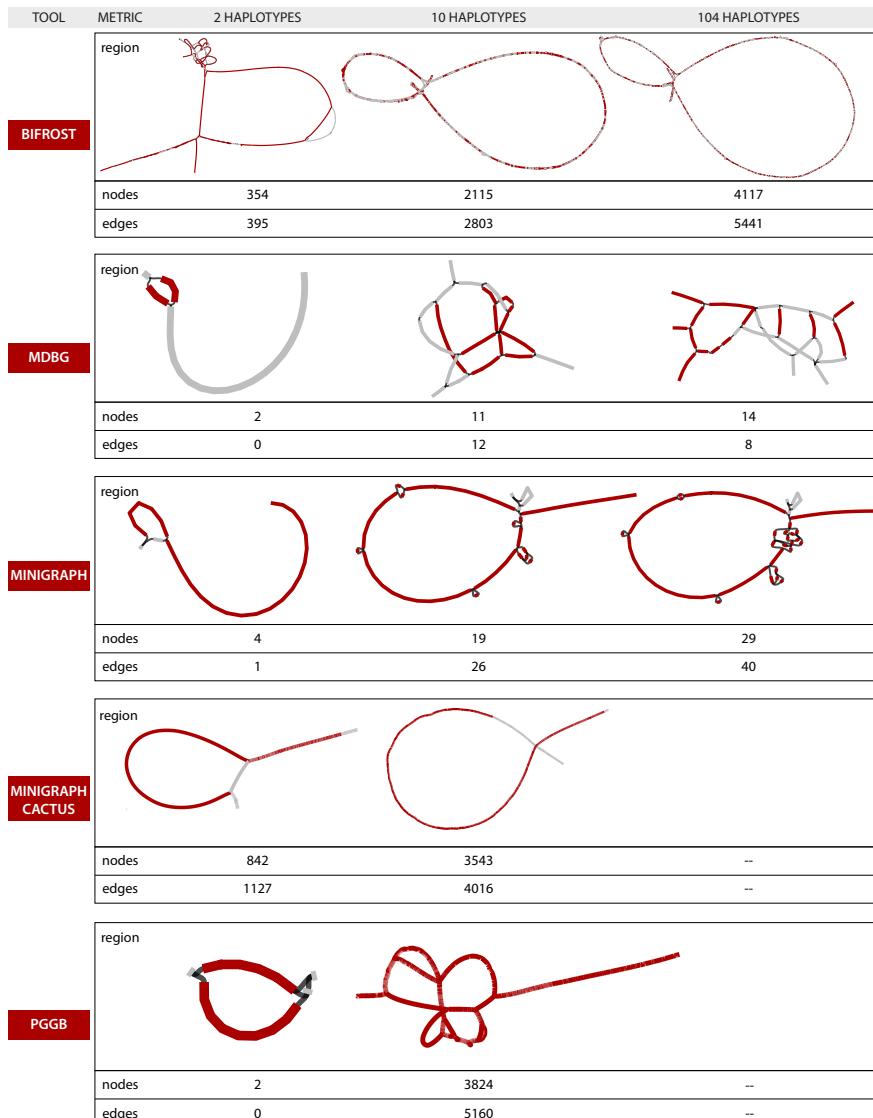


Figure 2.3: Representations of the complex HLA region by five graph construction methods over three increasing large human pangenomes. See caption of Fig. 2.2 for details.

2. Comparing methods for constructing and representing human pangenome graphs

algorithm. Yet in order to facilitate the reproducibility of results, pangenome building tools should generate an unchanged output from the same set of input sequences, independently of the particular run or the order in which these are given. We performed two tests to evaluate tool stability: i) we run the tools 3 times using as input the same H10 dataset and ii) we run the tools twice on shuffled input sequences, i.e. changing the order of the haplotypes of H10.

Bifrost and **mdbg** constructed exactly the same pangenome on every test, as by definition, De Bruijn graphs are stable. **Minigraph** generates identical graphs on identical inputs, but generates slightly different graphs when the input is permuted. Indeed the construction algorithm of **Minigraph** is order-sensitive as it augments the existing graph structure by aligning the next given haplotype to it and adding divergent sequences. **Minigraph-Cactus** generates slightly different graphs on identical input. **pggb** generated slightly different graphs while maintaining the same haplotype sequences in the paths. The overall representation of the input genomes is therefore preserved, while the topology of the variation graph varies. The first two of the three phases of the **pggb** pipeline (all-vs-all alignment and graph imputation) produce the same result on different runs with the same input but differences arise when the order of the input haplotypes changes. Most of the differences in the graph topology are thus due to the final smoothing steps.

Accessibility by downstream applications To facilitate their adoption, pangenome representations should be easily processed by downstream analyses. De Bruijn graphs are challenging to analyze due to their high number of nodes, edges, and redundancy (the $k - 1$ -overlaps between nodes). Though De Bruijn graph representations usually support queries of presence/absence on nodes (as in **Bifrost**), they lack tools able to perform more elaborate analyses such as those discussed in Section "Interpretation of variation in pangenome graphs: focus on two HLA loci", e.g. incorporating haplotype information at the k -mer level. On the other hand, variations graphs with paths provide more flexibility, i.e. as in **pggb** and **Minigraph-Cactus** with the **odgi** visualization toolkit. Finally in **Minigraph**, which considers a narrower spectrum of variants, the absence of path information prevents haplotype-level analysis; haplotypes would need to be manually mapped back to the graph. The choice of the pangenome building tool depends on the envisioned application. **pggb** and **Minigraph-Cactus** graphs have been shown to outperform linear references for short read mapping, genotyping and RNA sequencing mapping [8]. As these two methods are complex pipelines based on multiple tools where parameters have been carefully set, they can be more challenging to install and run than single integrated tools. **Minigraph** alone can also be used if the focus is on structural variation instead of SNPs or small indels, and to quickly produce a pangenome graph for complex loci visualization and interpretation. The dBG-based approaches show that, for example with **Bifrost**, they retain the same base-level information as more computational-heavy variation graph approaches, but the lack of tools to use them for analysis limits their adoption.

Haplotype compression Building a graph pangenome can be seen also as a way to store, compact and retrieve the input haplotypes. As the number of new assemblies is growing faster than the data storing capacity, pangenomes can potentially help save storage space. This is highlighted by the disk space reported in Table 2.1, which is consistently smaller than the sum of haplotype sizes for all methods and datasets.

In order to losslessly retrieve the input genomes from a pangenome, the representation has to store all variations from the original haplotype sequences as paths in the graph. **pggb** and **Minigraph-Cactus** fall into this category while the other three considered tools do not store paths, or do not consider all variations, thus they are lossy.

Of note, the GBZ tool [9] enables graph pangenomes that store paths in the GFA file format to be stored in a lossless compressed form. It uses a Graph Burrows-Wheeler transformation to compress the graph in a more efficient way than using gzip [9]. Using GBZ, the pangenomes generated by **pggb** and **Minigraph-Cactus** are losslessly compressed with space gains of 3.5-5x.

Ease of Visualization Visualizing large graphs which exceed hundreds of thousands of nodes is a challenge that exceeds the scope of pangenomics. The H104 pangenomes are difficult to visualize. Among the visualization tools considered by the Human PanGenome Reference consortium [6], only **Bandage** is able to display the **Minigraph** or **mdbg** H104 graphs, which contains a few million nodes. We reduced the number of nodes and edges of **pggb**, **Minigraph-Cactus** and **Bifrost** H10 graphs by collapsing isolated subgraphs representing SNPs or indels up to 10k bp (using the command `gfatools asm -b 10000 -u`).

Quality of Metadata and Annotation Augmenting pangenome structures with information from other omics data would increase pangenome relevance in biological discoveries. As biobanks are rapidly growing, more data is available on regulatory regions, transcriptomics, CNVs and other medically relevant traits [22, 23]. Pangenome data structures could leverage such information, and some of the considered tools offer basic functionality in this sense. **Bifrost** provides a function to link data to graph vertices through C++ APIs. **pggb** and **Minigraph-Cactus**, using **odgi**, offer annotation capabilities through insertion of paths or BED records. **Minigraph** and **mdbg** do not offer any annotation feature. Specifically, in order to enhance a pangenome graph with metadata (for example with genes and regulatory regions known variants), it is desirable to maintain compatibility with methods and data formats that use a linear reference. One could conceivably project data from a graph to a reference genome to continue downstream analyses using linear coordinates. A simple method to achieve this compatibility, in our view, is to store the reference genome of interest inside the graph pangenome that supports retrieving such a reference. Variation graphs built using **pggb** or **Minigraph-Cactus**, due to their locally acyclical and directed construction and their use of haplotype paths, store all the coordinates needed for such a task. Haplotype paths play an important role as

2. Comparing methods for constructing and representing human pangenome graphs

Table 2.2: **Relative strengths of five pangenome graph construction tools**

Explanation of rows: 1) efficacy of construction algorithm, measuring wall-clock time; 2) degree to which variants (e.g. SNPs) are retained; 3) ability of a tool to perform well on large datasets, both in comparison to other tools but also compared to smaller datasets; 4) ability to modify the produced data structure to add or remove haplotypes; 5) property of producing the same result irrespective of perturbations, such as permutation of the input order, and repeating the same run; 6) existence of tools (and operations) that can be applied to the resulting graphs; 7) whether input haplotypes information is retained by the tools, and if so, its space efficiency; 8) whether the entire graph can be directly visualized and interpreted; 9) easiness of ‘zooming in’ a specific genomic region and interpret variants; 10) summarizes the functionalities provided by the tools to annotate the pangenomes with genomic data; 11) ability to share information between the graph and a linear reference.

Metric	Bifrost	pggb	Minigraph-Cactus	Minigraph	mdbg
1) Construction speed	• • ○	• ○ ○	• ○ ○	• ○ ○	• ● ●
2) Variations	● ● ●	● ● ●	● ● ●	● ○ ○	● ○ ○
3) Scalability	● ● ●	● ○ ○	● ○ ○	● ○ ○	● ● ●
4) Editability	● ● ●	● ● ●	● ○ ○	● ○ ○	● ○ ○
5) Stability	● ● ●	● ○ ○	● ○ ○	● ○ ○	● ● ●
6) Accessibility by downstream applications	● ○ ○	● ● ●	● ● ●	● ○ ○	● ○ ○
7) Haplotype compression performance	● ○ ○	● ● ●	● ● ●	● ○ ○	● ○ ○
8) Ease of visualization	● ○ ○	● ● ○	● ○ ○	● ● ●	● ● ●
9) Loci visualization and interpretability	● ○ ○	● ● ○	● ● ●	● ○ ○	● ○ ○
10) Metadata and annotation	● ○ ○	● ● ●	● ○ ○	● ○ ○	● ○ ○
11) Compatibility with a linear reference coordinates	● ○ ○	● ● ●	● ● ●	● ○ ○	● ○ ○

they avoid additional mapping to the graph, by using the `odgitool` to extract or inject the required information. **Minigraph** does not store haplotype paths and requires mapping sequences to the graph to restore haplotype information. On the other hand, De Bruijn graphs, using associated color data, can record the membership of k-mers to a reference sequence, yet one cannot fully reconstruct a haplotype unless k-mers positions are also stored.

2.3 Discussion

Five state-of-the-art pangenome graphs construction tools were compared on the representation of up to 104 human haplotypes. The approaches significantly differ in terms of speed, graph size, and representation of variations. We find that it remains computationally prohibitive to generate human pangenome graphs for hundreds of haplotypes, especially while retaining all variations. Each approach has its own set of strengths, and ultimately the choice of the method depends on the downstream application. In addition, several takeaway points emerged from our analysis.

First, our focused analysis of HLA loci revealed that de Bruijn graphs and variation graphs represent genomic variations equally well as pangenomes. This is of particular importance as also shown by the draft human pangenome

references [8]: pangenomes are pivotal to trace complex and clinically relevant loci. While de Bruijn graphs are faster to construct, more stable, and scale better in terms of input size, the resulting graphs are challenging to interpret downstream. Variations graphs on the other hand are more practical to analyze at the expense of a less efficient construction step. Their visualization are more straightforward to interpret, mostly due to not having cycles, and provide insights into loci differences.

Second, we can highlight two categories of pangenomic methods that have distinct application domains. **pggb**, **Minigraph-Cactus** and **Bifrost** store all possible variations, and keep haplotype information as paths or colors. They provide a complete picture of the set of variations in the input genomes which makes them difficult to analyze. They can be used for a large variety of genomic analysis, as shown for **pggb** and **Minigraph-Cactus** [8]. **Minigraph** and **mdbq** generate 'sketched' pangenome graphs that consider only large variants, ignoring smaller differences, and are more efficient to construct and visualize. They can be used for large scale characterization of variation in population, as proven for bacteria [15].

Third, every tool possesses an exclusive set of features. **pggb** facilitates downstream analyses using the companion tool **odgi**. It allows to precisely extract and browse any locus of interest. It is the only tool that generates variation graphs without a reference. It also keeps a lossless representation of the input sequences. **Minigraph** generates a pangenome graph based on a reference sequence taken as a backbone. Its shines in the representation of complex structural variations, but does not include small or inter-chromosomal variations. The pipeline **Minigraph-Cactus**, that uses the **Cactus** base aligner, can be used to add small level variations on top of the **Minigraph** graph and to keep a lossless representation of the input sequences. **Bifrost** illustrates that classical de Bruijn graphs are scalable, stable, dynamic, and store all variations. However, extracting information from them remain a challenge. Lastly, **mdbq** is the fastest construction method which generates an approximate representation of differences between haplotypes. As discussed in Section "Accessibility by downstream applications", these features enable different genomic analyses and downstream applications.

2.4 Conclusions

In conclusion, our results highlight the strengths and weaknesses of current pangenome construction tools for human applications, with specific focus on how do they represent specific loci of medical relevance. We also provide insights on the features they possess and point out their best application domains. In our view, future directions for human pangenomes building tools should focus on tackling efficiency bottlenecks, aiming to represent hundreds to thousands of haplotypes. Representations should further be lossless and represent the input haplotypes as paths in the graph. Such features would unlock many other applications such as lossless compression of haplotypes and cancer copy number

2. Comparing methods for constructing and representing human pangenome graphs

variant analysis. Finally, we recognize the need for more user-friendly tools that can be used by biologists and that can translate complicated questions into graph queries. While `odgi` begins to address these questions in variation graphs, other approaches have not yet provided user-friendly interfaces. A package similar to `odgi` for de Bruijn graphs would help fully realize their potential.

2.5 Methods

Datasets and haplotypes collection

In order to evaluate the state of current human pangenome representations, we sought to build a human pangenome that contains all publicly available high-quality human haplotypes. We collected from two different sources 102 different haplotypes from the genome of 51 individuals, and also used the two reference genomes, GRCh38 from the Genome Reference Consortium (GRC) [24] and CHM13 v2.0 cell line of the T2T Consortium [25]. Five haplotypes correspond to Google Brain Genomics DeepConsensus [26] assembly dataset: they are hifiasm assemblies of PacBio Hi-Fi reads corrected with DeepConsensus. The average of their N50 is 37,99 Mbp. The remaining haplotype assemblies as well as the T2T reference are from the Human Pangenome Reference Consortium (HPRC) year-1 freeze [6], and GRCh38 is from the GRC. Their average N50 is 40.3 Mbp. Since HG002 is contained in the DeepConsensus data, the HPRC HG002 haplotypes were not used. The origin and the sex of the individuals are diverse and provide a fair representation of the diversity in human population: out of 51 total individuals, 21 are males and 30 are females and they represent 14 different ethnic groups, from US to Africa and Asia. We did not perform any additional selection, regarding sex and ethnicity, on these public datasets as our main goal was to use as many genomes as possible. However, the HPRC stated that the genomes were selected to represent genetic diversity in humans [8].

To evaluate the scalability of pangenome construction tools, we created three datasets of increasing size: 1) 2 haplotypes from the same individual, HG006, 2) 10 haplotypes from 5 different individuals (HG002, HG003, HG004, HG006 and HG00735) and finally 3) all of the 104 haplotypes. To test whether the order of the input sequences matters, we considered various random orderings for the 10 haplotypes in Dataset 2. Since `Minigraph` needs a reference sequence as first haplotype in order to correctly build the graph, we generated specific 2 and 10 haplotypes datasets with the first haplotype replaced by the reference genome CHM13. This was applied to the `Minigraph-Cactus` pipeline as well as it uses `Minigraph` variation graphs.

Pangenome graph construction tools

We evaluated tools that generate graph pangenomes as variation graphs and colored compacted de Bruijn graphs. Variation graphs are generally locally acyclic while de Bruijn graphs have cycles. In variation graphs, nodes represent sequences and edges represent immediate sequence adjacency without overlap.

Table 2.3: **Description of the three datasets generated to test the scalability of the tools**

Data sources: ¹ Google Brain Genomics [27]; ² Human Pangenome Reference Consortium [28]; ³ 1000 Genomes Project [28]; ⁴ Telomere to Telomere Consortium [28].

Haplotypes	Project	Bases
2	Google ¹	5.9 Gbp
10	Google, HPRC ²	30 Gbp
104	Google, HPRC, 1KG ³ , T2T ⁴	313.6 Gbp

Variation graphs are generally easier to visualize and to interpret while challenging to construct at scale and, apart from `pggb`, require a reference genome. In de Bruijn graphs (dBG), nodes are k -mers (string of length k) and edges are $(k-1)$ -overlaps between nodes. In practice, dBGs are represented in a compact way where all nodes along unbranching paths are compacted into *unitigs*. The resulting graph is called compacted De Bruijn Graph, where nodes are unitigs and edges represent $(k-1)$ -overlaps. As shown in Figure 2.1, de Bruijn graphs result in large graphs that pose visualization and interpretation challenges, in particular as there is no alignment to a reference.

- **Bifrost** constructs dynamic, coloured compacted de Bruijn Graphs (*cdBG*). It first generates a standard dBG using an efficient variant of Bloom Filters and then computes the compacted dBG from it. Colors, i.e. identifiers representing the sample origin of each k -mer are added by storing an array per k -mer. A human genome cdBG typically consists of a single large connected component, as common k -mers are shared between chromosomes. This pangenome representation contains all the variations present in input sequences.
- **mdbg** builds a variant of de Bruijn graphs called a minimizer-space de Bruijn Graph (**mdbg**), which is efficient to construct as it only considers a small fraction of the input nucleotides. Color information is currently not supported in the implementation. Similarly to Bifrost, a **mdbg** also typically represents a human genome as a single large connected component, albeit with orders of magnitude less nodes. Minimizer-space de Bruijn graphs mostly discard small variants, yet are sensitive to heterozygosity which creates branches in the graph.
- **Minigraph** constructs a directed, bidirected and acyclic variation graph iteratively by mapping new haplotypes using a combination of the minimap2 tool and the graph waveform alignment algorithm. The first input sequence acts as a backbone for the whole representation. The sample(s) of each node are stored in a rGFA output file. **Minigraph** considers only variations longer than 50 bps hence it is oblivious to isolated SNPs and small indels:

2. Comparing methods for constructing and representing human pangenome graphs

Table 2.4: URL, version, pangenome representation and parameters of the three analyzed tools.

pggb/0.2.0 used wfmash v0.7.0, seqwish v0.7.3 and smoothxg v0.6.1.

Tool	Github repository	Graph type	Version	Parameters
Bifrost	pmelest/Bifrost	De Bruijn graph	1.0.5	-k100 -c
pggb	pangenome/pggb	variation graph	0.2.0	-p 98 -s 10000 -k 311 -G 13033,13117 -O 0.03 -v -t 8 -T 8 -A -Z
Minigraph	lh3/Minigraph	variation graph	0.18	-cxggs
Minigraph-Cactus	ComparativeGenomics Toolkit/cactus	variation graph	2.2.3	-maxLen 10000 -delFilter 10000000
mdbg	ekimb/rust-mdbg	De Bruijn graph	1.0.1	-k 10 -d 0.0001 -minabund 1 -reference

even if it produces base-level alignment for contigs, the graphs are not a base-level resolution. The resulting graph is divided into connected components that correspond to the chromosomes present in the first given input genome.

- **Minigraph-Cactus** is a variation graph construction pipeline that combines **Minigraph** to generate a structural variations graph and **Cactus** base aligner to generate base-level pangenome graphs of a set of input assemblies and embg: The definition of as changed! Check your packages! Replacing it with the kernel definition on input line 145.ed haplotypes paths. **Cactus** [16] is a highly accurate and scalable reference-free multiple whole-genome alignment tool, that in this pipeline considers the reference sequence used by **Minigraph** to ensure that the resulting variation graph is acyclic. The final graph is further normalized using GFAffix[29]. The pipeline allows to generate multiple graphs, one for each chromosome, or produce a single graph that includes inter-chromosomal variants.
- **pggb** is a directed acyclic variation graph construction pipeline rather than a single tool. It calls three different tools: pairwise base-level alignment of haplotypes using wfmash [30], graph construction from the alignments with seqwish [31], graph sorting and normalization with smoothxg and GFAffix [32, 29]. The resulting variation graph represents variations of all lengths present in the input sequences.

Supplementary Information

Benchmark infrastructure

Running time of pangenome construction tools was measured as wall clock time and peak memory as maximum resident set size using the **time** command. Other metrics were obtained with custom Python scripts. All benchmarks were performed on a Supermicro Superserver SYS-2049U-TR4, with 3 TB RAM and 4 Intel SKL 6132 14-cores @ 2.6 GHz, using 8 cores.

TwoPaCo

We did not consider **TwoPaCo** as it is redundant with **Bifrost**. Both methods construct the same de Bruijn graphs. **TwoPaCo** is a method for constructing ccdBG by finding junction k -mers at the boundaries of unitigs or in branching nodes. It consists of two main steps in which it approximates the dBG with a Bloom filter in order to reduce the size of the problem and then runs a two pass highly parallel algorithm on it. It constructs ccdBGs similarly to **Bifrost**. **Bifrost** is faster, supports edit operations, and accepts also reads other than assemblies as input. We tested both tools on NCBI datasets from three different known human variation regions part of the human leukocyte antigen (HLA) complex: HLA-A, MICB and TAP1. These loci have different number of sequences and have complexity and length. The resulting graphs have exactly the same k -mer content and substantially equal topology. The difference is that **TwoPaCo** considers sequences with IUPAC 'N' bases while **Bifrost** does not and that in some cases **TwoPaCo** renders some unitigs split in two or more consecutive nodes.

Loci extraction method

For **Bifrost** and **mdbg** graphs, nodes corresponding to the input sequences are identified with **GraphAligner** [33] and the subgraph is extracted using the **Bandage** *reduce* function. As the aligned nodes are not expected to represent the full diversity of the population in the pangenomes, the considered portion of the graph contains also nodes up to a certain distance from the aligned ones: 1 for **mdbg** and 3 for **Bifrost**. This number is based on the size of the sequences spelled by the nodes and on the considered variations. Artifacts, mostly tips, that are not part of the locus of interest are removed with a custom python script. For **Minigraph** generated graphs, the **Minigraph** own alignment function has been used to identify the nodes and then **Bandage** is used to extract the subgraph. For **pggb**, first we generate a bed file of the position of the region of interest in every haplotype used to construct the graph. The ranges are derived from aligning them to the locus sequence(s) using minimap2 [34]. The graph corresponding to the region is then extracted using the **odgi** extract and **odgi** view functions. For **Minigraph-Cactus** we use the same strategy as **pggb**, with the difference that the bed file is only for the reference CHM13, present in the graph.

The annotation of the specific loci in the subgraph is done using nodes from the alignment with **Minigraph** or **GraphAligner** to the subgraph. This makes it possible to highlight multiple sections in the region, as, for example, genes and pseudogenes of interest.

Availability of data and materials

The scripts used to generate and analyse the pangenomes can be found at [35][36] under MIT license. Google Brain Genomic assemblies can be found at [27].

2. Comparing methods for constructing and representing human pangenome graphs

HPRC assemblies, CHM13 and GRCh38 can be found at [28].

Funding

R.C was supported by ANR Full-RNA, SeqDigger, Inception and PRAIRIE grants (ANR-22-CE45-0007, ANR-19-CE45-0008, PIA/ANR16-CONV-0005, ANR-19-P3IA-0001). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grants agreements No. 872539 and 956229.

Author's contributions

FA, YD and RC conceived and designed the project. FA implemented the scripts. FA and PL ran the experiments. FA, YD, PL and RC wrote the paper. The authors read and approved the final manuscript.

Series of dissertations submitted to the Institut Pasteur Paris, Université de Paris Cité, University of Oslo No. 1234 ISSN 1234-5678 All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission

Authors' affiliations

Francesco Andreace Institut Pasteur, Université Paris Cité, G5 Sequence Bioinformatics
Sorbonne Université, Collège doctoral
25-28 Rue du Dr Roux, 75015 Paris

Pierre Lechat Bioinformatics and Biostatistics Hub, Institut Pasteur, Université de Paris Cité
25-28 Rue du Dr Roux, 75015 Paris

Yoann Dufresne Institut Pasteur, Université Paris Cité, G5 Sequence Bioinformatics
Bioinformatics and Biostatistics Hub, Institut Pasteur, Université de Paris Cité
25-28 Rue du Dr Roux, 75015 Paris

Rayan Chikhi Institut Pasteur, Université Paris Cité, G5 Sequence Bioinformatics
25-28 Rue du Dr Roux, 75015 Paris

2.6 Perspectives

- In perspective the work of this article could have been extended to provide a pipeline (on nextflow for example) to benchmark all pangnomes constructed

independently of the data structure used to provide the features described on the paper. This would have provided a platform to independently benchmark new pangenomes and rendered the paper not only useful as review and hints for end-users but also as benchmarking for new methods or methods improvement;

- Among the features I could have included it would have been useful:
 1. count SNPs (trivial for VGs, not trivial for DBGs);
 2. node length distribution;
 3. node degree distribution;
 4. conversion of variation graphs to De Bruijn graphs (extending nodes smaller than k) to validate if they contain the same information/ did not lose information - see **Minigraph** example;

2. Comparing methods for constructing and representing human pangenome graphs

2.7 Building a *Lodderomyces elongisporus* pangenome reference: overcoming current limitations.

Here we present another example of pushing the boundaries of variation graph pangenome building strategies, precisely in producing a graph that can represent inter-chromosomal events, like rearrangements, for the medically interesting yeast strain of *Lodderomyces elongisporus*. We show also how a pangenome based approach can help produce a more comprehensive insight of such events than a linear genome based one.

The work presented here is also in part a team effort with other PhD students and researcher I was co-leading with Daniel Doerr during a winter-school organized by my consortium: Simon Heumos, which I would like to thank for the discussion we had on the construction of such graphs and Nicola Rizzo. Most of the results and findings reported here are mostly my own, subsequent, work.

The following sections will explain the process needed to produce custom, biologically-significant and biologically-driven pangenome graphs of a yeast strain with the current state-of-the-art tools and offer insights on workarounds that can be used in cases with special needs, where current tools features limit the analysis that can be done. This will also show how pangenome models can offer more powerful tools to analyze specific variation in a group of similar genomes compared to linear reference based ones.

In this specific case, the goal was to produce a variation graph that considered, and showed, the long chromosomal rearrangements that were noticed by the team that produced the assemblies from short and long reads. Chromosome-crossing syntenies for multiple contigs in alignment hits of chromosomes C, G, and H, were detected, suggesting that they could correspond to a singular same recombination group. As this finding was consistent with an independent genomic study, based on SNPs, performed on the isolates of a fungemia outbreak in a neonatal intensive care unit in Delhi between 2021 and 2022. This study also noticed that this rearrangements are more frequent in hospital and patient associated populations than in fruit ones [37].

2.7.1 *Lodderomyces elongisporus*: genetic characteristichs, interest and used data

Lodderomyces elongisporus is a diploid yeast that has been isolated from, among many sources, humans and it is recently emerging as pathogenic. It is phylogenetically placed in the Candida clade and the size of its genome is usually between 15 and 16 Mb, 2 orders of magnitude smaller than a human genome [38]. Its DNA is organized into 8 chromosomes, that here will be referred in alphabetical order and decreasing size A to H, from around 3.5Mbp of chr A to 800 Kbp of chr H, plus a 35 Kbp mitochondrial DNA. Our analysis shows that it has a stable core genome of 13Mbp. Increasing reports of (mostly bloodstream) infection in mainly immunosuppressed adults makes it an increasingly important subject of studies: an outbreak war reported occurring in a neonatal ICU in

Dheli, India from September 2021 to February 2022 with 1 death[37]. Samples sequenced by us were called in alphabetical order, followed with a number greater or equal than 0 that denoted the quality of the assembly. Moreover, one sample, B2, for which a member of the consortium produced a high quality assembly after manual curation, was used as relative reference in the cohort. Another sample, J2, was also fully resolved into chromosomes while the others were assembled into contigs.

2.7.2 Building ad hoc pangenome reference

We produced a dBG of the assemblies using **Bifrost**, but their usefulness remains limited for visual analysis of complex biological events interpretation and study.

Figure 2.4a shows the visualization of the dBG for k -mer length equal to 25: redundant parts of the genome make the graph collapse. This is the same phenomenon previously described for human genomes, as better insight can be acquired when just a small region is visualized. De Bruijn Graphs can instead be useful for to produce quite straightforward whole-genome alignment- and reference-free phylogenetic analysis in a fraction of time required by competitor methods that use all vs. all alignment. The tool **SANS serif** [39] can process directly a **Bifrost** generated cdBG to estimate the phylogenetic splits between the genomes contained in the graph. Figure 2.4b shows the visualization of the phylogenetic network produced by **SANS serif** using the tool **SplitsTree** [40]. This results shows how dBGs are quite powerful when applied to specific applications.

Given high quality assemblies generated by the sequences of these 11 samples, we decided to also build a pangenome graph with small variant resolution using **pggb** and **Minigraph-Cactus**, in a similar way to what has been done with the Human Draft Pangenome Reference [8]. It is important to notice that in order to produce the best biological correct result, several rounds of parameter tuning and manual curation are needed, with knowledge far superior of the one of a first-time user.

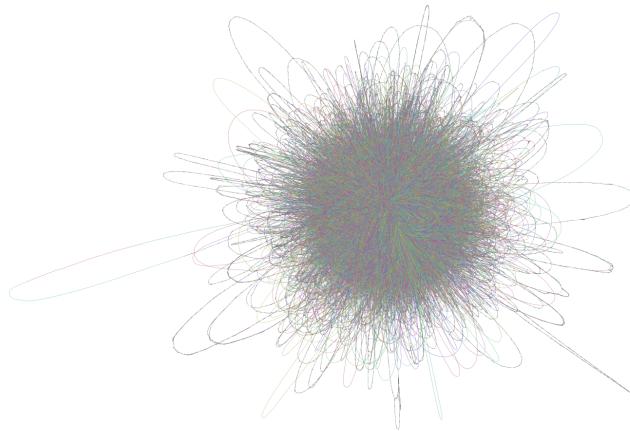
The first step to build such pangomes is to divide the genome assemblies into communities of sequences belonging to chromosomes.

2.7.2.1 Determining chromosomal communities

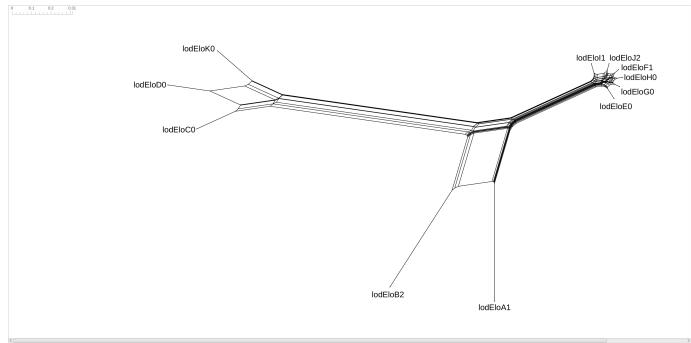
Variation graphs construction pipelines use mapping or alignment between the input set of genomes to infer graphs. Their first step consists in grouping the sequences from the assemblies into communities representing a chromosome in order to run a single computation instance and produce a separate graph for each of them. The final graph is then given by joining together the output of each group. This means that without any pre-processing, no inter-chromosomal event can be detected.

In this specific use-case, in order to identify inter-chromosomal events, contigs associated to any of the 3 chromosomes conjectured to be part of the

2. Comparing methods for constructing and representing human pangenome graphs



(a) **Bandage** visualization of the pangenome dBG of the cohort of 11 *Lodderomyces elongisporus* strains. As for human genomes, visualization of the whole data offers no particular insight.



(b) **SplitsTree** visualization of the phylogeny network generated using **SANS serif** from the cdBG constructed using **Bifrost**. Image obtained by modifying a pipeline developed in collaboration with Nicola Rizzo.

Figure 2.4: Visualization of the cdBG representation and phylogeny analysis of the *Lodderomyces elongisporus* pangenome.

rearrangement had to fall into the same community and be provided together in input to the pipeline. This would ensure that, if such rearrangement exists, it would produce a feature in the graph that would show as a tangle between the chromosomes.

As the rest of the assemblies, with the exception of the J2 sample, were not resolved into single chromosomes, each of them was aligned to the reference B2 using **wfmash** alignment segment size of 10k and 95% sequence identity (and lower segment size, 90% sequence identity if unmapped). The identity scores of

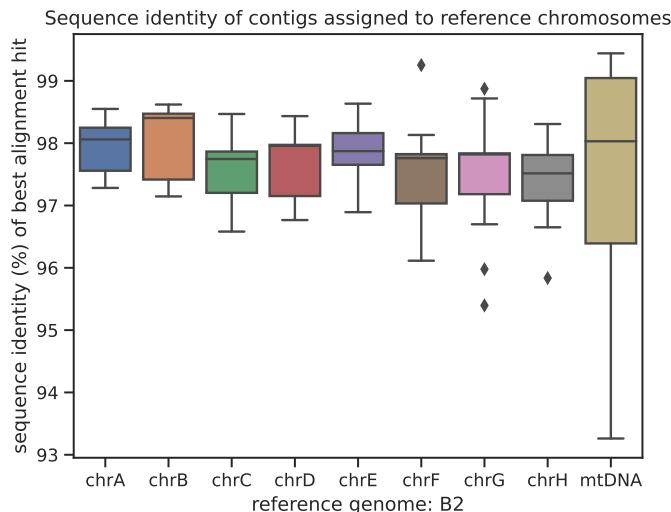


Figure 2.5: Sequence Identity of contigs assigned to reference chromosomes. Image produced using a pipeline developed by Simon Heumos.

the alignment of the genomes to the reference B2 assembly is shown in figure 2.5. From this alignment, contigs were assigned to the chromosomal community to which they best mapped. Finally, the ones of chromosome C, G and H were grouped into a single one.

2.7.2.2 Producing a variation graph using pggb

As pggb uses all-vs-all alignment of a collection of sequence as first step to infer the graph, it enables the representation of recombination among chromosomes placed inside the same community, as seen also for human acrocentric chromosomes [41].

By simply running the `nextflow/pangenome (pggb)` pipeline, we were able to produce a first pangenome representation of the 11 yeast strains. The tangle visible in figure AAA clearly shows the recombination happening between the three chromosomes. This work was mainly done by Simon Heumos and the graph shown in figure AAA is the result of more than 25 rounds of parameters tuning. The detection of the event with a variation graph using pggb encouraged the effort to produce a similar representation with `Minigraph-Cactus`, a pipeline that is not designed to construct grouped chromosomes pangenomes.

2.7.2.3 Overcoming Minigraph-Cactus limitation by modifying both the data and the pipeline

In order to produce a graph that represents variation between multiple chromosome with `Minigraph-Cactus`, a custom pipeline has to be used.

2. Comparing methods for constructing and representing human pangenome graphs

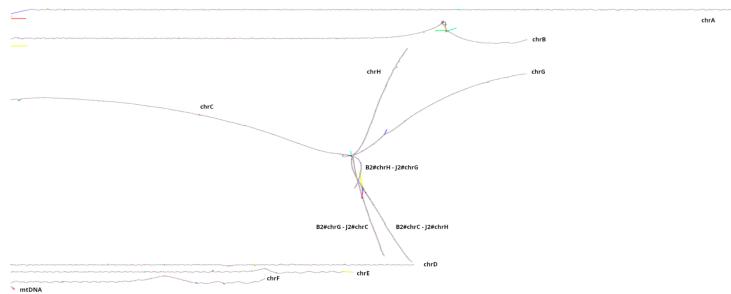


Figure 2.6: Gfaestus visualization of the `pggb` variation graph of the 11 *Lodderomyces elongisporus* samples. Image produced by Simon Heumas.

Minigraph-Cactus communities are implicitly inferred by the first step, performed by **Minigraph**. For this tool, the chromosomes present in the first reference genome given as input are used as communities and backbone for the whole graph and no sequence can be both assigned to different chromosome. This is an intrinsic characteristic of **Minigraph** and cannot be changed with input parameters: it means that there is no feature to have chromosome C,G and H considered together in input.

To try to overcome this limitation of the approach we tried to produce a graph that respected the condition of having the three chromosome inside the same connected component, at the cost of producing a representation that was not biologically correct. We therefore produced a chimeric contig consisting of the concatenation of the three chromosomes assemblies of the B2 sample. The rationale was to provide the 3 chromosomes chained together as a single backbone in the **Minigraph** construction step. This would allow sequences to be mapped to any of chr C, G and H to be considered together in the subsequent steps of alignment and graph the **Minigraph-Cactus** pipeline. The expectation was to therefore produce a graph that showed the recombination from the mapping of the contigs of the other genomes.

By building a graph using **Minigraph** with the chimeric chromosome CGH and all the contigs of the other genomes assigned to chromosome C, G and H does not represent any recombination event, as can be seen in figure 2.7a. This was somewhat expected, as it is known that **Minigraph** does not also consider inversion between genomes. When the complete modified pipeline of **Minigraph-Cactus** is run, it is possible to see the tangle between the chromosomes, as shown in figures 2.7b 2.10.

2.7.3 Representing recombination events from groups of genomes

Applying simple community separation on the all vs all alignment of the contigs, like the one suggested in the manual of `pggb`, does not help confirming the hypothesis, mainly because of segment length selection. Figure 2.8 shows



(a) Graph of chimeric chromosome CGH from sample B2 and all the contigs of the other genomes aligning to it produced with **Minigraph**. The graph is linear and no inter-chromosomal event is visible.



(b) The graph after all the other steps of the **Minigraph-Cactus** pipeline, colored by depth, after simplification of variants < 1kbp using the command `gfatools asm -b 1000 -u`. The large recombination event is now visible.

Figure 2.7: Difference in output between **Minigraph** and **Minigraph-Cactus** of the chimeric graph produced to visualize the inter-chromosomal event between C,G and H.

community detection using Louvain algorithm on the contig network inferred by all vs all alignment. This inter-chromosomal rearrangement is instead detectable using linear whole-genome assembly based tools. By aligning J2 to the relative reference genomes B2 with **wfmash** and then looking for syntenies and rearrangements with **SyRI**, it is possible to detect syntenic path (longest set of co-linear regions), structural rearrangements (inversions, translocations, and duplications) [42]. Figure 2.9 shows the detected rearrangements and duplications between the 3 chromosomes using **plotsr** [43]. While figure 2.9 shows in a clear way the kind of inter-chromosomal variation between the 2 strains, **SyRI** does work only with genomes resolved to chromosome level. This means that such analysis is not possible on the whole cohort using standard linear reference tools. It is therefore possible to visualize the tangle that emerges from the graph generated with **Minigraph-Cactus**.

2.8 Conclusion and Perspectives

2. Comparing methods for constructing and representing human pangenome graphs

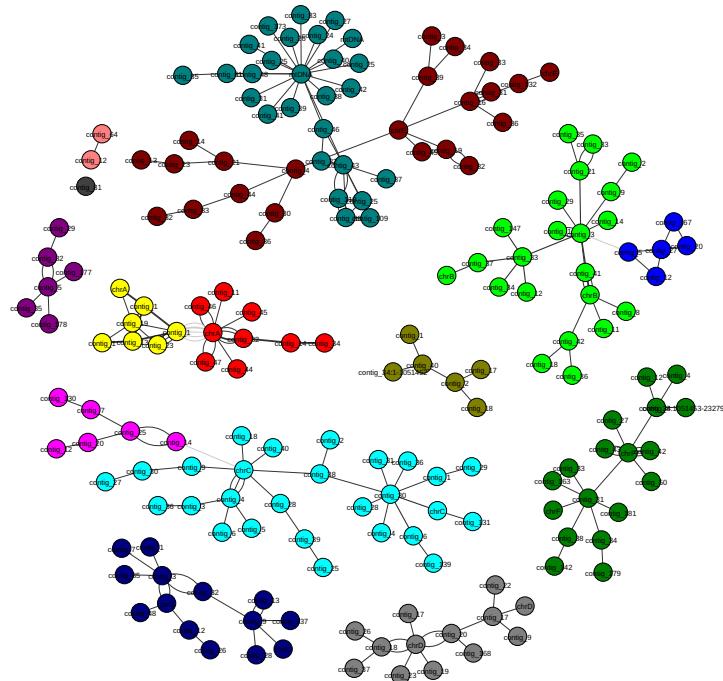


Figure 2.8: Community partition of the contigs based on alignment scores.

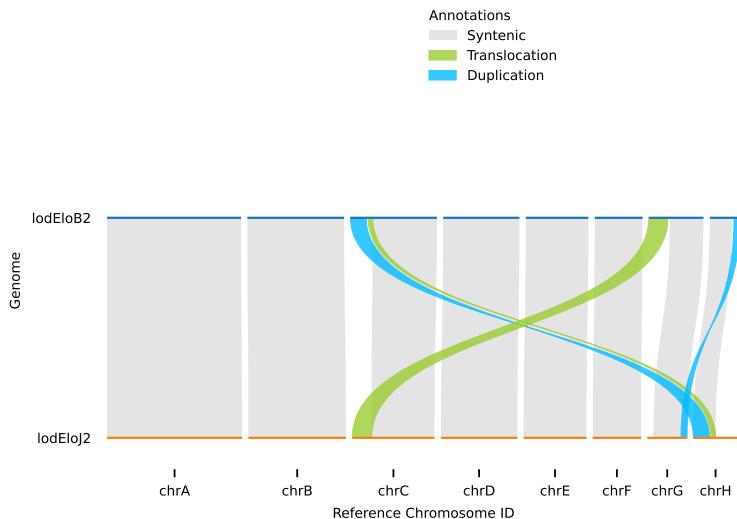


Figure 2.9: `plotsr` visualization of the inter-chromosomal recombination detected using `wfmash` and `SyRI`.

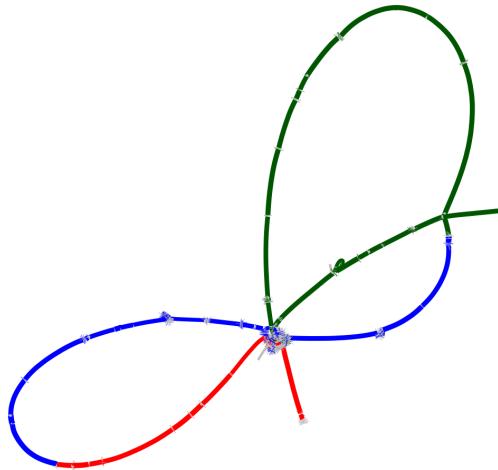
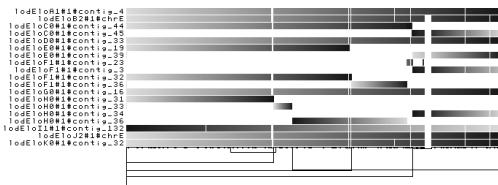
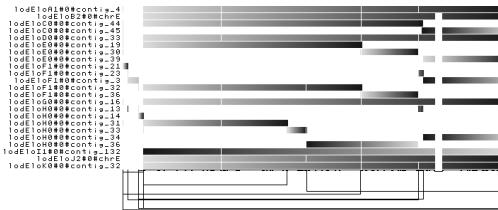


Figure 2.10: The tangle of chromosomes C, G and H in the **Minigraph-Cactus** variation graph. Nodes are colored based on **Minigraph** alignment of chromosome C (dark green), G (blue) and H (red) of the reference assembly B2. The three chromosomes are bound together because of the construction.



(a) One dimensional visualization of the variation graph built with **pggb**, containing 19 contigs from the assemblies, selected before construction using all vs all alignment data.



(b) One dimensional visualization of the variation graph built with **Minigraph-Cactus**, containing 23 contigs from the assemblies, selected automatically by the pipeline.

Figure 2.11: One dimensional visualization of chromosome E variation graphs of **pggb** and **Minigraph-Cactus** using **odgi**. Differences can easily be noted in the leftmost parts of the images.

Bibliography

- [1] Haeussler, M., Zweig, A. S., Tyner, C., Speir, M. L., Rosenbloom, K. R., Raney, B. J., Lee, C. M., Lee, B. T., Hinrichs, A. S., Gonzalez, J. N., et al. “The UCSC genome browser database: 2019 update”. In: *Nucleic acids research* vol. 47, no. D1 (2019), pp. D853–D858.
- [2] Garrison, E. et al. “Variation graph toolkit improves read mapping by representing genetic variation in the reference”. In: *Nature Biotechnology* vol. 36, no. 9 (Oct. 2018), pp. 875–879.
- [3] Consortium, T. C. P.-G. “Computational pan-genomics: status, promises and challenges”. In: *Briefings in Bioinformatics* vol. 19, no. 1 (Oct. 2016), pp. 118–135. ISSN: 1477-4054. DOI: 10.1093/bib/bbw089. eprint: <https://academic.oup.com/bib/article-pdf/19/1/118/25406834/bbw089.pdf>. URL: <https://doi.org/10.1093/bib/bbw089>.
- [4] Sirén, J. et al. “Pangenomics enables genotyping of known structural variants in 5202 diverse genomes”. In: *Science* vol. 374, no. 6574 (2021), abg8871. DOI: 10.1126/science.abg8871. eprint: <https://www.science.org/doi/pdf/10.1126/science.abg8871>. URL: <https://www.science.org/doi/abs/10.1126/science.abg8871>.
- [5] Sherman R.M., S. S. “Pan-genomics in the human genome era.” In: *Nat Rev Genet*, no. 21 (2020), pp. 243–254. DOI: <https://doi.org/10.1038/s41576-020-0210-7>.
- [6] Wang, T. et al. “The Human Pangenome Project: a global resource to map genomic diversity”. In: *Nature* vol. 604, no. 7906 (Apr. 2022), pp. 437–446. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04601-8. URL: <https://doi.org/10.1038/s41586-022-04601-8>.
- [7] Ebler, J. et al. “Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes”. In: *Nature Genetics* vol. 54, no. 4 (Apr. 2022), pp. 518–525. ISSN: 1546-1718. DOI: 10.1038/s41588-022-01043-w. URL: <https://doi.org/10.1038/s41588-022-01043-w>.
- [8] Liao, W.-W. et al. “A draft human pangenome reference”. In: *Nature* vol. 617, no. 7960 (May 2023), pp. 312–324. ISSN: 1476-4687. DOI: 10.1038/s41586-023-05896-x. URL: <https://doi.org/10.1038/s41586-023-05896-x>.
- [9] Sirén, J. and Paten, B. “GBZ file format for pangenome graphs”. In: *Bioinformatics* vol. 38, no. 22 (Sept. 2022), pp. 5012–5018. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac656. eprint: <https://academic.oup.com/bioinformatics/article-pdf/38/22/5012/47153721/btac656.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btac656>.

Bibliography

- [10] Sheikhzadeh, S., Schranz, M. E., Akdel, M., Ridder, D. de, and Smit, S. “PanTools: representation, storage and exploration of pan-genomic data”. In: *Bioinformatics* vol. 32, no. 17 (Sept. 2016), pp. i487–i493.
- [11] Holley G., M. P. “Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs.” In: *Genome Biol*, no. 21 (2020), p. 249. DOI: <https://doi.org/10.1186/s13059-020-02135-8>.
- [12] Garrison, E. et al. “Building pangenome graphs”. In: *bioRxiv* (2023). DOI: [10.1101/2023.04.05.535718](https://doi.org/10.1101/2023.04.05.535718). eprint: [https://www.biorxiv.org/content/early/2023/04/06/2023.04.05.535718](https://www.biorxiv.org/content/early/2023/04/06/2023.04.05.535718.full.pdf). URL: <https://www.biorxiv.org/content/early/2023/04/06/2023.04.05.535718>.
- [13] Minkin, I., Pham, S., and Medvedev, P. “TwoPaCo: an efficient algorithm to build the compacted de Bruijn graph from many complete genomes”. In: *Bioinformatics* vol. 33, no. 24 (Sept. 2016), pp. 4024–4032. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btw609](https://doi.org/10.1093/bioinformatics/btw609). eprint: <https://academic.oup.com/bioinformatics/article-pdf/33/24/4024/25168506/btw609.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btw609>.
- [14] Li, H., Feng, X., and Chu, C. “The design and construction of reference pangenome graphs with minigraph.” In: *Genome Biol*, no. 21 (2020), p. 265. DOI: <https://doi.org/10.1186/s13059-020-02168-z>.
- [15] Ekim, B., Berger, B., and Chikhi, R. “Minimizer-space de Bruijn graphs: Whole-genome assembly of long reads in minutes on a personal computer”. In: *Cell Systems* vol. 12, no. 10 (2021), 958–968.e6. ISSN: 2405-4712. DOI: <https://doi.org/10.1016/j.cels.2021.08.009>. URL: <https://www.sciencedirect.com/science/article/pii/S240547122100332X>.
- [16] Armstrong, J. et al. “Progressive Cactus is a multiple-genome aligner for the thousand-genome era”. In: *Nature* vol. 587, no. 7833 (Nov. 2020), pp. 246–251. ISSN: 1476-4687. DOI: [10.1038/s41586-020-2871-y](https://doi.org/10.1038/s41586-020-2871-y). URL: <https://doi.org/10.1038/s41586-020-2871-y>.
- [17] Hickey, G. et al. “Pangenome graph construction from genome alignments with Minigraph-Cactus”. In: *Nature Biotechnology* (May 2023). ISSN: 1546-1696. DOI: [10.1038/s41587-023-01793-w](https://doi.org/10.1038/s41587-023-01793-w). URL: <https://doi.org/10.1038/s41587-023-01793-w>.
- [18] Chin, C.-S., Behera, S., Metcalf, G., Gibbs, R. A., Boerwinkle, E., and Sedlazeck, F. J. “A pan-genome approach to decipher variants in the highly complex tandem repeat of LPA”. In: *bioRxiv* (2022). DOI: [10.1101/2022.06.08.495395](https://doi.org/10.1101/2022.06.08.495395). eprint: [https://www.biorxiv.org/content/early/2022/06/10/2022.06.08.495395](https://www.biorxiv.org/content/early/2022/06/10/2022.06.08.495395.full.pdf). URL: <https://www.biorxiv.org/content/early/2022/06/10/2022.06.08.495395>.
- [19] Dendrou, C. A., Petersen, J., Rossjohn, J., and Fugger, L. “HLA variation and disease”. In: *Nature Reviews Immunology* vol. 18, no. 5 (May 2018), pp. 325–339. ISSN: 1474-1741. DOI: [10.1038/nri.2017.143](https://doi.org/10.1038/nri.2017.143). URL: <https://doi.org/10.1038/nri.2017.143>.

- [20] Vietzen, H., Zoufaly, A., and Traugott, M. e. a. “Deletion of the NKG2C receptor encoding KLRC2 gene and HLA-E variants are risk factors for severe COVID-19.” In: *Genet Med* vol. 23 (2021), pp. 963–967. DOI: 10.1038/s41436-020-01077-7.
- [21] Guerracino, A., Heumos, S., Nahnsen, S., Prins, P., and Garrison, E. “ODGI: understanding pangenome graphs”. In: *Bioinformatics* (May 2022). btac308. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac308. eprint: <https://academic.oup.com/bioinformatics/advance-article-pdf/doi/10.1093/bioinformatics/btac308/43882774/btac308.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btac308>.
- [22] “100,000 Genomes Pilot on Rare-Disease Diagnosis in Health Care — Preliminary Report”. In: *New England Journal of Medicine* vol. 385, no. 20 (2021). PMID: 34758253, pp. 1868–1880. DOI: 10.1056/NEJMoa2035790. eprint: <https://doi.org/10.1056/NEJMoa2035790>. URL: <https://doi.org/10.1056/NEJMoa2035790>.
- [23] Johnson, R. et al. “Leveraging genomic diversity for discovery in an electronic health record linked biobank: the UCLA ATLAS Community Health Initiative”. In: *Genome Medicine* vol. 14, no. 1 (Sept. 2022), p. 104. ISSN: 1756-994X. DOI: 10.1186/s13073-022-01106-x. URL: <https://doi.org/10.1186/s13073-022-01106-x>.
- [24] Schneider, V. A. et al. “Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly”. In: *Genome Res* vol. 27, no. 5 (Apr. 2017), pp. 849–864.
- [25] Nurk, S. et al. “The complete sequence of a human genome”. In: *Science* vol. 376, no. 6588 (2022), pp. 44–53. DOI: 10.1126/science.abj6987. eprint: <https://www.science.org/doi/pdf/10.1126/science.abj6987>. URL: <https://www.science.org/doi/abs/10.1126/science.abj6987>.
- [26] Baid, G. et al. “DeepConsensus improves the accuracy of sequences with a gap-aware sequence transformer”. In: *Nature Biotechnology* (Sept. 2022). ISSN: 1546-1696. DOI: 10.1038/s41587-022-01435-7. URL: <https://doi.org/10.1038/s41587-022-01435-7>.
- [27] Baid, G. et al. *Dataset. Google Brain Assemblies*. 2023. URL: https://console.cloud.google.com/storage/browser/brain-genomics-public/research/deepconsensus/publication/analysis/genome_assembly.
- [28] Liao, W.-W. et al. *Dataset. Human Pangenome Reference Consortium Assemblies*. 2023. URL: <https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=working/>.
- [29] Doerr, D. “Gfaffix identifies walk-preserving shared affixes in variation graphs and collapses them into a non-redundant graph structure.” In: (2021). URL: <https://github.com/marschall-lab/GFAffix>.
- [30] Guerracino, A., Mwaniki, N., Marco-Sola, S., and Garrison, E. *wfmash: whole-chromosome pairwise alignment using the hierarchical wavefront algorithm*. Sept. 2021. URL: <https://github.com/ekg/wfmash>.

Bibliography

- [31] Garrison, E. and Guarracino, A. “Unbiased pangenome graphs”. In: *Bioinformatics* vol. 39, no. 1 (Nov. 2022), btac743. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btac743. eprint: <https://academic.oup.com/bioinformatics/article-pdf/39/1/btac743/48448986/btac743.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btac743>.
- [32] Guarracino, A. and Garrison, E. *smoothxg: local reconstruction of variation graphs using partial order alignment*. 2021. URL: <https://github.com/pangenome/smoothxg>.
- [33] Rautiainen, M. and Marschall, T. “GraphAligner: rapid and versatile sequence-to-graph alignment”. In: *Genome Biology* vol. 21, no. 1 (Sept. 2020), p. 253. ISSN: 1474-760X. DOI: 10.1186/s13059-020-02157-2. URL: <https://doi.org/10.1186/s13059-020-02157-2>.
- [34] Li, H. “Minimap2: pairwise alignment for nucleotide sequences”. In: *Bioinformatics* vol. 34, no. 18 (May 2018), pp. 3094–3100. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty191. eprint: <https://academic.oup.com/bioinformatics/article-pdf/34/18/3094/25731859/bty191.pdf>. URL: <https://doi.org/10.1093/bioinformatics/bty191>.
- [35] Andreace, F. *Github sourcecode*. 2023. URL: <https://github.com/frankandreace/CRHPG>.
- [36] Andreace, F. *Zenodo sourcecode*. 2023. DOI: 10.5281/zenodo.8370336.
- [37] Yadav, A., Jain, P., Jain, K., Wang, Y., Singh, A., Singh, A., Xu, J., and Chowdhary, A. “Genomic Analyses of a Fungemia Outbreak Caused by Lodderomyces elongisporus in a Neonatal Intensive Care Unit in Delhi, India”. In: *mBio* vol. 14, no. 3 (2023), e00636–23. DOI: 10.1128/mbio.00636-23. eprint: <https://journals.asm.org/doi/pdf/10.1128/mbio.00636-23>. URL: <https://journals.asm.org/doi/abs/10.1128/mbio.00636-23>.
- [38] Wang, Y. and Xu, J. “Lodderomyces elongisporus: An emerging human fungal pathogen”. In: *PLOS Pathogens* vol. 19, no. 9 (Sept. 2023), pp. 1–9. DOI: 10.1371/journal.ppat.1011613. URL: <https://doi.org/10.1371/journal.ppat.1011613>.
- [39] Rempel, A. and Wittler, R. “SANS serif: alignment-free, whole-genome-based phylogenetic reconstruction”. In: *Bioinformatics* vol. 37, no. 24 (June 2021), pp. 4868–4870. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab444. eprint: <https://academic.oup.com/bioinformatics/article-pdf/37/24/4868/50334826/btab444.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btab444>.
- [40] Huson, D. H. and Bryant, D. “The SplitsTree App: interactive analysis and visualization using phylogenetic trees and networks”. In: *Nature Methods* (Sept. 2024). ISSN: 1548-7105. DOI: 10.1038/s41592-024-02406-3. URL: <https://doi.org/10.1038/s41592-024-02406-3>.

- [41] Guaracino, A. et al. “Recombination between heterologous human acrocentric chromosomes”. In: *Nature* vol. 617, no. 7960 (May 2023), pp. 335–343. ISSN: 1476-4687. DOI: 10.1038/s41586-023-05976-y. URL: <https://doi.org/10.1038/s41586-023-05976-y>.
- [42] Goel, M., Sun, H., Jiao, W.-B., and Schneeberger, K. “SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies”. In: *Genome Biology* vol. 20, no. 1 (Dec. 2019), p. 277. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1911-0. URL: <https://doi.org/10.1186/s13059-019-1911-0>.
- [43] Goel, M. and Schneeberger, K. “plotsr: visualizing structural similarities and rearrangements between multiple genomes”. In: *Bioinformatics* vol. 38, no. 10 (Apr. 2022), pp. 2922–2926. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac196. eprint: <https://academic.oup.com/bioinformatics/article-pdf/38/10/2922/49010748/btac196.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btac196>.

Papers

Appendices