Journal
doi.10.1093/xxxxxx/yyyyyy
Advance Access Publication Date: Day Month Year
Genome Analysis

Genome Analysis

Cluster-efficient pangenome graph construction with nf-core/pangenome

Simon Heumos [1,2,3,4,*], Michael L. Heuer [5], Friederike Hanssen [1,2,3,4], Lukas Heumos [6,7,8], Andrea Guarracino [9,10,], Peter Heringer [1,2,3,4], Philipp Ehmele [6], Pjotr Prins [9], Erik Garrison [9], Sven Nahnsen [1,2,3,4,*]

[1]Quantitative Biology Center (QBiC) Tübingen, University of Tübingen, Tübingen, Germany
[2]Biomedical Data Science, Dept. of Computer Science, University of Tübingen, Tübingen, Germany
[3]M3 Research Center, University Hospital Tübingen, Tübingen, Germany
[4]Institute for Bioinformatics and Medical Informatics (IBMI), Eberhard-Karls University of Tübingen, Tübingen, Germany
[5]University of California, Berkeley, Berkeley, 94720, California, USA
[6]Institute of Computational Biology, Department of Computational Health, Helmholtz Munich, Germany
[7]Comprehensive Pneumology Center with the CPC-M bioArchive, Helmholtz Zentrum Munich, Member of the German Center for Lung Research (DZL), Munich, Germany
[8]TUM School of Life Sciences Weihenstephan, Technical University of Munich, Freising, Germany
[9]Department of Genetics, Genomics and Informatics, University of Tennessee Health Science Center, 71 S Manassas St, Memphis, 38163, Tennessee, USA
[10]Human Technopole, Viale Rita Levi-Montalcini 1, 20157, Milan, Italy

[*]To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Pangenome graphs offer a comprehensive way of capturing genomic variability across multiple genomes. However, current construction methods often introduce biases, excluding complex sequences or relying on references. The PanGenome Graph Builder (PGGB) addresses these issues. To date, though, there is no state-of-the-art pipeline allowing for easy deployment, efficient and dynamic use of available resources, and scalable usage at the same time.
Results: To overcome these limitations, we present *nf-core/pangenome*, a reference-unbiased approach implemented in Nextflow following nf-core's best practices. Leveraging biocontainers ensures portability and seamless deployment in HPC environments. Unlike PGGB, nf-core/pangenome distributes alignments across cluster nodes, enabling scalability. Demonstrating its efficiency, we constructed pangenome graphs for 1000 human chromosome 19 haplotypes and 2146 *E. coli* sequences, achieving a two to threefold speedup compared to PGGB without increasing greenhouse gas emissions.
Availability: nf-core/pangenome is released under the MIT open-source license, available on GitHub and Zenodo, with documentation accessible at https://nf-co.re/pangenome/1.1.2/docs/usage.
Supplementary: Supplementary data are available at *Bioinformatics* online.
Contact: simon.heumos@qbic.uni-tuebingen.de, sven.nahnsen@qbic.uni-tuebingen.de

1 Introduction

The availability of high-quality population-wide whole-genome assemblies (Liao *et al.*, 2023; Kang *et al.*, 2023; Weller *et al.*, 2023; Zhou *et al.*, 2022; Liu *et al.*, 2020; Leonard *et al.*, 2022) offers new opportunities to study sequence evolution and variation within and between genomic populations. A challenge is simultaneously representing and analyzing hundreds to thousands of genomes at a gigabase scale. One solution here is a pangenome. It models a population's entire set of genomic sequences (Ballouz *et al.*, 2019). In contrast to reference-based genomic approaches, which relate sequences to a linear genome, pangenomics relates each new sequence to all the others represented in the pangenome (The Computational Pan-Genomics Consortium, 2016; Eizenga *et al.*, 2020; Sherman and Salzberg, 2020) minimizing reference-bias. Pangenomes can be described as sequence graphs which store DNA sequences in nodes with edges connecting the nodes as they occur in the individual sequences (Hein, 1989). Genomes are encoded as paths traversing the nodes (Garrison *et al.*, 2018).

Current pangenome graph construction methods exclude complex sequences or are reference-biased (Chin *et al.*, 2023; Minkin *et al.*, 2016). One recent approach that overcomes such limitations is the PanGenome Graph Builder (PGGB) pipeline (Garrison *et al.*, 2023). PGGB iteratively refines an all-to-all whole-genome alignment graph that lets us explore sequence conservation and variation, infer phylogeny, and identify recombination events. PGGB has been extensively evaluated (Garrison *et al.*, 2023; Andreace *et al.*, 2023) and applied to build the first draft human pangenome reference (Liao *et al.*, 2023). However, PGGB is implemented in bash, which (a) makes it difficult to deploy on HPC systems, (b) does not allow for a fine granular tuning of computing resources for different steps of the pipeline (Sztuka *et al.*, 2024), and (c) limits its cluster scalability to one node. These limitations greatly hinder the broad application of large-scale pangenomes.

To compensate for that, we wrote *nf-core/pangenome*, a reference-unbiased approach to construct pangenome graphs. Mirroring PGGB, nf-core/pangenome is implemented in Nextflow (Di Tommaso *et al.*, 2017). In contrast to PGGB, nf-core/pangenome can distribute the quadratic all-to-all base-level alignments across nodes of a cluster by splitting the approximate alignments into problems of equal size. We benchmarked the time spent on base-pair level alignments and show that it is reduced linearly with an increase in alignment problem chunks (Suppl. 5.5). We showcase the workflow's scalability by applying it to 1000 chromosome 19 human haplotypes and 2146 *E. coli* sequences, which were built in less than half the time PGGB required while not increasing the CO2 equivalent (CO2e) emissions.

2 Material and Methods

2.1 Pipeline overview

The pipeline's (Fig. 1a) input is a FASTA file compressed with *bgzip* (Li *et al.*, 2009) containing the sequences to create the graph. Sequence names should follow the Pangenome Sequence Naming specification (PanSN-spec) (https://github.com/pangenome/PanSN-spec, last accessed October 2024). The primary output is a pangenome variation graph (Garrison *et al.*, 2018) in the Graphical Fragment Assembly (GFA) format version 1 (http://gfa-spec.github.io/GFA-spec/GFA1.html, last accessed October 2024).

2.1.1 Core workflow

The core workflow of nf-core/pangenomemirrors PGGB (Fig. 1a) with additional enhancements: (a) All concurrent processes can be run in parallel. (b) Each process can be given individual computing resources.

The pipeline begins with an all-to-all alignment of the input sequences using the whole-chromosome pairwise sequence aligner WFMASH (https://github.com/waveygang/wfmash, last accessed October 2024), avoiding reference, order, or orientation bias, allowing every sequence to serve as a reference. In the pangenome graph induction step SEQWISH (Garrison and Guarracino, 2022), an alignment to variation graph inducer, converts the sequence alignments into a variation graph. The graph is then simplified using SMOOTHXG (Garrison *et al.*, 2023): A 1-dimensional (1D) graph embedding (Heumos *et al.*, 2024) orders the graph's nodes to best match the nucleotide distances of the genomic paths of the graph. Next, the graph is split into partially overlapping segments. The sequences of each segment are realigned with a local Multiple Sequence Alignment (MSA) kernel, partial order alignment (POA) (Lee *et al.*, 2002). Afterwards, the segments are laced back together into a variation graph. By default, the SMOOTHXG process is applied 3 times in order to smooth the edge effects at the boundaries of the segments. Finally, we employ GFAFFIX (Liao *et al.*, 2023) to systematically condense redundant nodes within the graph.

Graph quality is assessed with ODGI (Guarracino *et al.*, 2022), which provides statistics and visualizations. Optionally, variants can be called against any (reference) path(s) in the graph using *vg deconstruct* (Garrison *et al.*, 2018). Results are summarized in a MultiQC (Ewels *et al.*, 2016) report. Pipeline implementation details are given in Suppl. 5.1.

If desired, the pipeline performs community detection to identify clusters of related sequences in the pangenome graph, revealing biological patterns such as conserved or divergent regions across genomes (Supplementary 5.2), with the core workflow executed for each community in parallel.

3 Results

3.1 Building a 1000 haplotypes chr19 pangenome graph

The Human Pangenome Resource Consortium (HPRC) recently built a draft human pangenome of 90 haplotypes. However, haplotype data for thousands of individuals was already generated by the 1000 Genomes Project (1KGP) (Durbin *et al.*, 2010). As a use case, we used nf-core/pangenome to build a graph of 1000 chromosome 19 haplotypes (Kuhnle *et al.*, 2020) in 3 days, emitting 51.07 kg CO2e. PGGB took 7 days for the same task (56.32 kg CO2e). In Fig. 1b the pangenome growth curve generated with PANACUS (Liao *et al.*, 2023) shows nucleotide growth as more haplotypes are added. The softcore pangenome, defined as sequences traversed by 95% of haplotypes, comprises the majority of the pangenome even with 1000 haplotypes. This stability may be due to the exclusion of complex regions like the centromere in the short-read data.

3.2 Building a 2146 sequences *E. coli* pangenome graph

To evaluate the pipeline's scalability, we built a pangenome graph of 2146 *E. coli* sequences. The nf-core/pangenome graph was completed in 10 days, emitting 175.18 kg CO2e, while PGGB could not finish within 30 days due to cluster time restrictions. For the growth curve (Fig. 1c) we excluded 130 plasmid sequences. The softcore pangenome remains stable at ∼3Mb, but its size constitutes less than 10% of the total pangenome. This substantial pangenomic growth is likely driven by horizontal gene transfer, as bacteria incorporate genes from one another at various genomic locations. Other reasons could be sequencing errors or human contamination (Breitwieser *et al.*, 2019).

4 Discussion

We implemented nf-core/pangenome, an easy-to-install, portable, and cluster-scalable pipeline for unbiased pangenome variation graph construction. It is the first pangenomic pipeline within the nf-core framework that enables the comparative analysis of gigabase-scale pangenome datasets. While tools like Minigraph (Li *et al.*, 2020) or PGR-TK (Chin *et al.*, 2023) also address pangenome analysis, nf-core/pangenome uniquely integrates this capability into the standardized nf-core framework, offering compatibility with a wide range of modular workflows and community-developed best practices.

The pipeline's core workflow has been successfully applied to *Neisseria meningitidis* (Yang *et al.*, 2023), wild grapes (Cochetel *et al.*, 2023), humans (Guarracino *et al.*, 2023; Liao *et al.*, 2023), grapevines (Guo *et al.*, 2024), taurines (Milia *et al.*, 2024), and rats (Villani *et al.*, 2024) underpinning the community effort to focus on a best-practice workflow to create reference-unbiased and sequence complete pangenome graphs. The modular domain-specific language (DSL) 2 pipeline structure facilitates easy exchange of processes with alternative tools, expanding its functionality and integration with other (sub-)workflows.

We have shown that we are able to perform all-vs-all base pair level alignments of thousands of sequences. When executed on an HPC,

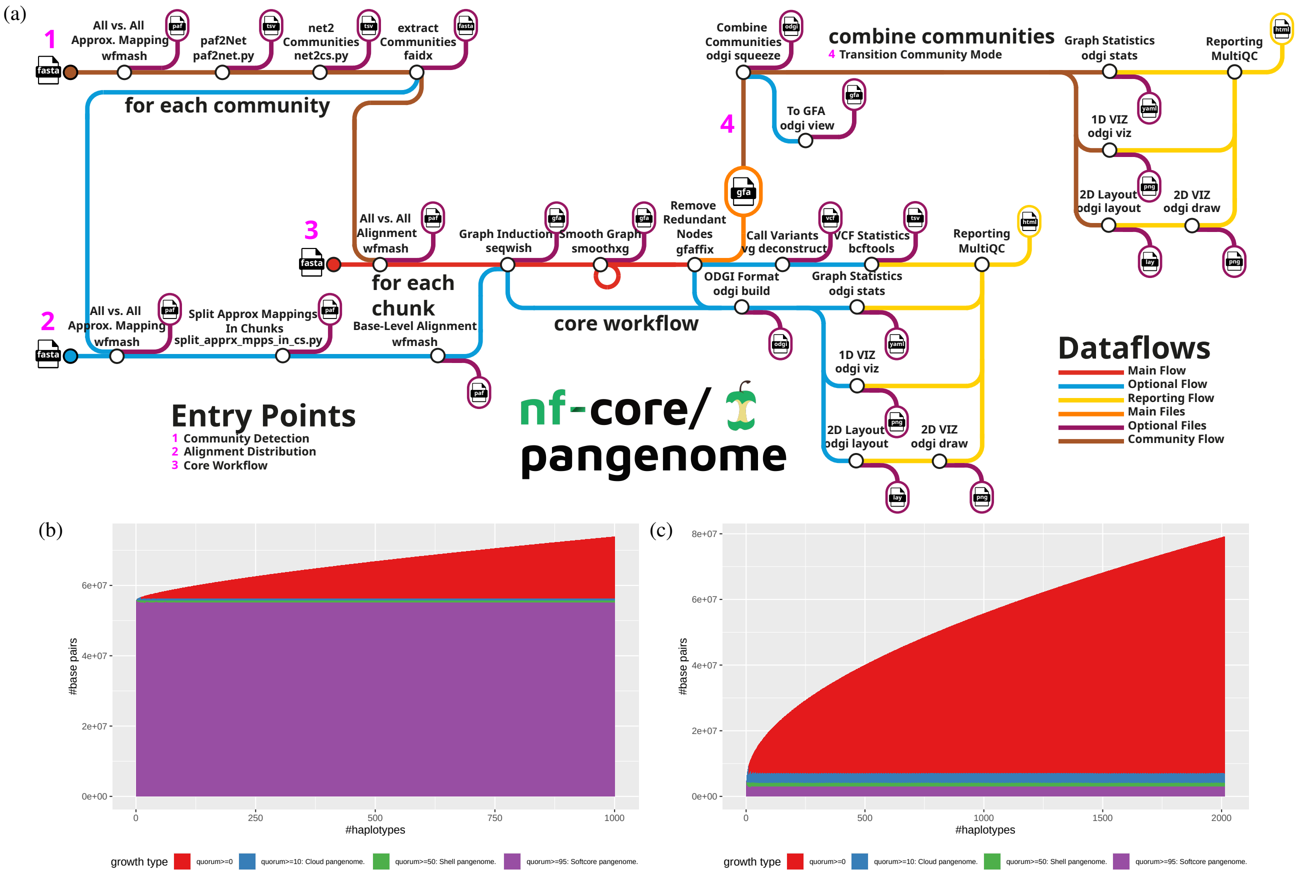


Fig. 1. (a) Schematic representation of the nf-core/pangenome workflow processes and detailed analysis steps. The input consists of one FASTA file containing all sequences. The pipeline comes with 3 major entry points: (1) Community detection, which identifies clusters of related sequences or regions in the pangenome graph to reveal biologically significant patterns like conserved or divergent areas across genomes (Supplementary 5.2), (2) alignment distribution, and (3) core workflow. Optional community detection (1) is performed on the input sequences. If selected, the heavy all-to-all baise-pair level alignments (2) can be split into problems of equal size. nf-core/pangenome's core workflow (3) is a direct mirror of PGGB. If running in community mode, all communal graphs are combined into one (4) and the subsequent quality control subworkflow is executed. The output is a pangenome graph in GFA format. (b) + (c) Pangenome growth curves of the built pangenome graphs. Growth type is defined as the minimum fraction of haplotypes that must share a graph feature after each time a haplotype is added to the growth histograph. $quorum >= 0$: All sequences without any filtering are considered. $quorum >= 10$: Sequences traversed by at least 10% of the haplotypes. $quorum >= 50$: Sequences traversed by at least 50% of haplotypes. $quorum >= 95$: Sequences traversed by 95% of haplotypes. (b) Pangenome growth curve of the chromosome 19 pangenome graph of 1000 haplotypes. (c) Pangenome growth curve of the E. coli pangenome graph of 2013 haplotypes.

nf-core/pangenome's parallel workflow accelerates graph construction compared to PGGB. PGGB's inability to assign individual computational resources to each pipeline step leads to the allocation of one whole node of an HPC, despite the fact that some processes can only make use of one thread. This blocks valuable CPU cycles. In contrast, nf-core/pangenome leverages Nextflow's process management for optimal resource allocation, crucial for cloud-based executions.

Competing pipelines either lack workflow management system (Chin *et al.*, 2023), or their workflow language of choice is e.g. Toil (Vivian *et al.*, 2017; Hickey *et al.*, 2023) which makes them less user-friendly, less cluster-efficient, and less portable (Wratten *et al.*, 2021). nf-core/pangenome is currently the only pangenomics pipeline that is optionally monitoring its CO2 footprint. The measurements have shown that constructing extensive pangenome graphs, such as the 2146 *E. coli* graph, requires a considerable amount of energy. Therefore, we recommend assessing the rationale and methodology before conducting energy-intensive experiments.

Although we expect our pipeline to scale for future challenges, such as for the next HPRC phase which targets 350 individuals, further optimizations are possible: The IMplicit Pangenome Graph (IMPG) (https://github.com/ekg/impg, last accessed October 2024) tool extracts homologous loci from genomes mapped to a specific target region. This would allow us to break the whole genome multiple alignments into smaller pieces, construct a pangenome graph for each piece, and lace these together into a full graph with gfalace (https://github.com/pangenome/gfalace, last accessed October 2024).

We anticipate the pipeline, or its parts, will enhance current single linear reference analysis methods to explore whole population variation instead of focusing on one reference only. Looking ahead, pangenome construction pipelines like nf-core/pangenome will play a pivotal role in studying entire populations, single-cell whole genome sequencing analysis, and constructing personalized (medical) pangenome references (Sirén *et al.*, 2024).

Software and data availability

Code and data resources for this manuscript and its figures are available in the public repository: https://github.com/subwaystation/pangenome-paper.

Acknowledgments

We thank Matthias Seybold from QBiC for maintaining the Core Facility Cluster. We thank Sabrina Krakau from QBiC for giving feedback to the nf-co2footprint plugin section. We are grateful to the nf-core community for their support during the implementation of the pipeline. From the nf-core community, we want to thank Matthias Hörtenhuber, Maxime Garcia, Susanne Jodoin, Julia Mir Petrol, Adam Talbot, and Gisela Gabernet.

Funding

S.H. acknowledges funding from the Central Innovation Programme (ZIM) for SMEs of the Federal Ministry for Economic Affairs and Energy of Germany. This work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A). A.G. acknowledges support from the Human Technopole. S.N. acknowledges support from iFIT funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2180—390900677 and CMFI under EXC 2124–390838134. We gratefully acknowledge support from NIH/NIDA U01DA047638 (E.G.), NIH/NIGMS R01GM123489 (E.G., P.P.), and NSF PPoSS Award #2118709 (E.G., P.P.), and the CITG (E.G.).

Competing interests

Author L.H. is employed by LaminLabs.

References

Andreace, F. *et al.* (2023). Comparing methods for constructing and representing human pangenome graphs. *Genome Biology*, **24**(1).

Ballouz, S. *et al.* (2019). Is it time to change the reference genome? *Genome Biology*, **20**(1), 159.

Breitwieser, F. P. *et al.* (2019). Human contamination in bacterial genomes has created thousands of spurious proteins. *Genome Research*, **29**(6), 954–960.

Chin, C.-S. *et al.* (2023). Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. *Nature Methods*, **20**(8), 1213–1221.

Cochetel, N. *et al.* (2023). A super-pangenome of the North American wild grape species. *Genome Biology*, **24**(1).

Di Tommaso, P. *et al.* (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*, **35**(4), 316–319.

Durbin, R. M. *et al.* (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061–1073.

Eizenga, J. M. *et al.* (2020). 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. and Guarracino, A. (2022). Unbiased pangenome graphs. *Bioinformatics*, **39**(1).

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.* (2023). Building pangenome graphs. *bioRxiv*.

Guarracino, A. *et al.* (2022). ODGI: understanding pangenome graphs. *Bioinformatics*, **38**(13), 3319–3326.

Guarracino, A. *et al.* (2023). Recombination between heterologous human acrocentric chromosomes. *Nature*, **617**(7960), 335–343.

Guo, L. *et al.* (2024). Super Pangenome of Grapevines Empowers Improvement of the Oldest Domesticated Fruit. *bioRxiv*.

Hein, J. (1989). A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. *Molecular Biology and Evolution*.

Heumos, S. *et al.* (2024). Pangenome graph layout by path-guided stochastic gradient descent. *Bioinformatics*, **40**(7).

Hickey, G. *et al.* (2023). Pangenome graph construction from genome alignments with minigraph-cactus. *Nature Biotechnology*, **42**(4), 663–673.

Kang, M. *et al.* (2023). The pan-genome and local adaptation of arabidopsis thaliana. *Nature Communications*, **14**(1).

Kuhnle, A. *et al.* (2020). Efficient construction of a complete index for pan-genomics read alignment. *Journal of Computational Biology*, **27**(4), 500–513.

Lannelongue, L. *et al.* (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, **8**(12).

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

Leonard, A. S. *et al.* (2022). Structural variant-based pangenome construction has low sensitivity to variability of haplotype-resolved bovine assemblies. *Nature Communications*, **13**(1).

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.

Liao, W.-W. *et al.* (2023). A draft human pangenome reference. *Nature*, **617**(7960), 312–324.

Liu, Y. *et al.* (2020). Pan-genome of wild and cultivated soybeans. *Cell*, **182**(1), 162–176.e13.

Milia, S. *et al.* (2024). Taurine pangenome uncovers a segmental duplication upstream of *KIT* associated with depigmentation in white-headed cattle. *bioRxiv*.

Minkin, I. *et al.* (2016). TwoPaCo: an efficient algorithm to build the compacted de Bruijn graph from many complete genomes. *Bioinformatics*, **33**(24), 4024–4032.

Sayers, E. W. *et al.* (2021). Database resources of the national center for biotechnology information. *Nucleic Acids Research*, **50**(D1), D20–D26.

Sherman, R. M. and Salzberg, S. L. (2020). Pan-genomics in the human genome era. *Nature Reviews Genetics*, **21**(4), 243–254.

Sirén, J. *et al.* (2024). Personalized pangenome references. *Nature Methods*.

Sztuka, M. *et al.* (2024). Nextflow vs. plain bash: different approaches to the parallelization of SNP calling from the whole genome sequence data. *NAR Genomics and Bioinformatics*, **6**(2), lqae040.

The Computational Pan-Genomics Consortium (2016). Computational pan-genomics: status, promises and challenges. *Briefings in Bioinformatics*, page bbw089.

Traag, V. A. *et al.* (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, **9**(1).

Villani, F. *et al.* (2024). Pangenome reconstruction in rats enhances genotype-phenotype mapping and novel variant discovery. *bioRxiv*.

Vivian, J. *et al.* (2017). Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology*, **35**(4), 314–316.

Weller, C. A. *et al.* (2023). Highly complete long-read genomes reveal pangenomic variation underlying yeast phenotypic diversity. *Genome Research*, **33**(5), 729–740.

Wratten, L. *et al.* (2021). Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nature Methods*, **18**(10), 1161–1168.

Yang, Z. *et al.* (2023). Pangenome graphs in infectious disease: a comprehensive genetic variation analysis of neisseria meningitidis leveraging oxford nanopore long reads. *Frontiers in Genetics*, **14**.

Zhou, Y. *et al.* (2022). Assembly of a pangenome for global cattle reveals missing sequences and novel structural variations, providing new insights into their diversity and evolutionary history. *Genome Research*, **32**(8), 1585–1601.

5 Supplement

5.1 Implementation

nf-core/pangenome is written in Nextflow using its latest domain-specific language (DSL) 2 syntax which facilitates a modular pipeline structure. Each software tool is an individual process that is implemented in its own module (https://nf-co.re/docs/contributing/modules, last accessed October 2024). Processes are concatenated into subworkflows (https://nf-co.re/docs/contributing/subworkflows, last accessed October 2024). Developed with the nf-core framework, the pipeline follows a set of best-practice guidelines ensuring high-quality development, maintenance, and testing standards. Specifically, we provide community support via a dedicated Slack channel (https://nfcore.slack.com/channels/pangenome, last accessed October 2024), GitHub issues, and detailed documentation (https://nf-co.re/pangenome/1.1.2/docs/usage, last accessed October 2024).

Versioning and portability are enabled through (a) semantic versioning (https://semver.org/, last accessed October 2024) of the pipeline via tagged releases on GitHub, (b) packaging software dependencies in archivable containers so that the software compute environment is the same across different systems, and (c) summarizing software versions and parameters in the MultiQC report of the pipeline. nf-core/ pangenome uses biocontainers to facilitate portability across different computing resources like HPC clusters, cloud platforms, or local machines. Code changes are evaluated with GitHub Actions' continuous integration (CI) using a pipeline-specific small test data set. For each new pipeline release, a full-size test is run on Amazon Web Services (AWS) validating the code integrity and cloud compatibility of real-world data sets. Specifically, a pangenome graph is created from the 8 Saccharomyces cerevisiae strains of the Yeast Population Reference Panel (YPRP) (Yue and Liti 2018). The results of such a run are available on the nf-core webpage (https://nf-co.re/pangenome/1.1.2/results/pangenome/results-0e8a38734ea3c0397f94416a0146a2972fe2db8b, last accessed October 2024). Because we implemented our processes using DSL2 nf-core/modules (https://github.com/nf-core/modules, last accessed October 2024), they can be distributed easily to other users to share commonly used processes or subworkflows across pipelines. This boosts the reuse of existing work done by the community to be integrated into future pipelines.

5.2 Chromosome community detection

Chromosome community detection involves identifying clusters of closely related sequences or genomic regions within the pangenome graph. These communities represent areas of high similarity or shared sequence variants across multiple genomes. Detecting such communities allows researchers to focus on biologically meaningful patterns, such as conserved or highly divergent regions, which are crucial for understanding genetic diversity, evolutionary relationships, and genotype-phenotype associations.

For example, eukaryotic genomes are typically organized into chromosomes, and this organization is considered during graph construction in community detection mode. Chromosome groupings from the input sequence are examined, and homologies detected in the all-to-all WFMASH mapping step are passed through the Leiden clustering algorithm (Traag *et al.*, 2019), where the edge weight is calculated as $mapped\_length * mapped\_identity$. The nf-core/pangenome core workflow is then executed in parallel for each of the resulting communities. The communal graphs are subsequently merged into a single graph, followed by a final round of quality control (see Fig 1a, brown tubes).

This approach works effectively for large input sequences with a mapping length filter greater than 1 Mb, as demonstrated by Guarracino *et al.* (2023) when exploring recombination between heterologous human acrocentric chromosomes.

5.3 Compute environment

We applied the nf-core/pangenome pipeline v1.1.2 to various inputs evaluating both the scalability of the all-vs-all alignment step as well as the pipeline as a whole. We used Nextflow version 23.10.1.5891 and Singularity version 3.8.7-1.el8 for each pipeline run. Experiments were conducted on our core facility cluster (CFC) with 24 Regular nodes (32 cores / 64 threads with two AMD EPYC 7343 processors with 512 GB RAM and 2 TB scratch space) and 4 HighMem nodes (64 cores / 128 threads with two AMD EPYC 7513 processors with 2048 GB RAM and 4TB scratch space). Each Nextflow process was given at most 64 threads. This ensures a fair run time comparison with PGGB v0.5.4 which was always executed on one Regular node via Slurm.

5.4 Estimation of the carbon footprint of pipeline runs

We also estimated the carbon dioxide equivalent (CO2e) emissions of each nf-core/pangenome pipeline run using the nf-co2footprint Nextflow plugin (https://github.com/nextflow-io/nf-co2footprint, last accessed October 2024) v1.0.0-beta. Using the Nextflow resource usage metrics and information about the power consumption of the compute system, the plugin first estimates the energy consumption for each pipeline task. It then uses the consumed energy's location-specific carbon intensity to estimate the respective CO2e emission. The calculations are based on the carbon footprint computation method developed in the Green Algorithms project (www.green-algorithms.org) (Lannelongue *et al.*, 2021).

5.5 Alignment jobs distribution

The computationally heavy all versus all base-pair level alignments can be distributed across nodes of a cluster: First, WFMASH is run in mapping mode (WFMASH MAP), finding all sequence homologies using approximate alignments. The resulting Pairwise mApping Format (PAF) file is split into chunks of equal problem size. The number of chunks is manually selected. The value can be guided by the number and size of the input sequences, and by the available hardware. Assuming the number of chunks equals the number of nodes on a cluster, then potentially each base-pair level alignment (WFMASH ALIGN) can be run in parallel on each node (Fig. 1a, cyan tubes). All resulting PAFs are then forwarded to the pipeline's core workflow which is continued at the SEQWISH process.

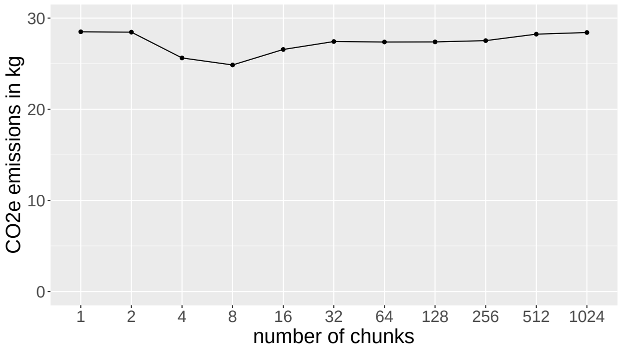
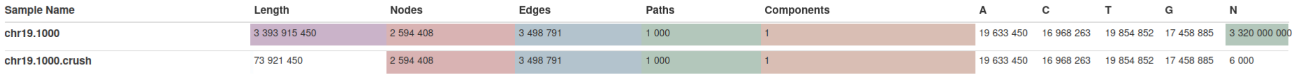


Fig. S1. Base-pair level alignment evaluation. CO2e emissions are stable across varying numbers of chunks.

5.6 1KGP chromosome 19 data set

The FASTA of the chromosome 19 data set was downloaded in December 2023 from http://dolomit.cs.tu-dortmund.de/chr19.1000.fa.xz. The data



Sample Name	Length	Nodes	Edges	Paths	Components	A	C	T	G	N
chr19.1000	3 393 915 450	2 594 408	3 498 791	1 000	1	19 633 450	16 968 263	19 854 852	17 458 885	3 320 000 000
chr19.1000.crush	73 921 450	2 594 408	3 498 791	1 000	1	19 633 450	16 968 263	19 854 852	17 458 885	6 000

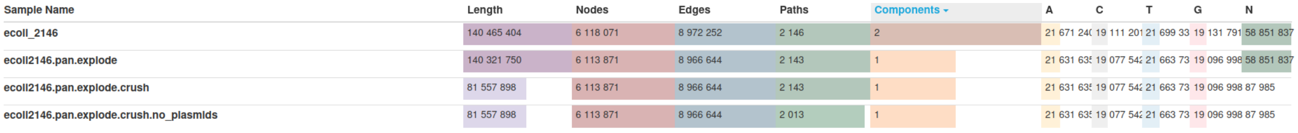
Fig. S2. Screenshot of the output of ODGI’s MultiQC module displaying the vital graph statistics calculated by odgi stats of the 1000 Genomes Project 1000 haplotypes chromosome 19 pangenome graphs. In the crushed graph consecutive Ns of all nodes containing Ns were merged into just one N per node. A: Number of adenine bases in the graph. C: Number of cytosine bases in the graph. T: Number of thymine bases in the graph. G: Number of guanine bases in the graph. N: Number of bases with unknown base identity.

Fig. S3. odgi draw 2D layout displaying the graph topology of the crushed 1KGP pangenome graph. Structural variation would appear as bubbles.

set is described in (Kuhnle *et al.*, 2020). Statistics of the built pangenome graph can be seen in Supplementary Fig. 5.6. The initial graph contains over 97% of Ns. We applied *odgi crush* (odgi version 0.8.6), which crushes consecutive Ns of all nodes containing Ns into just one N per node, to the 1KGP chromosome 19 pangenome graph. This brings down the number of Ns from over 3B to exactly 6000 (Suppl. Fig. 5.6). The 2D visualization (Suppl. Fig. 5.6) is perfectly linear without any large SVs present hinting that only short-read data was used to create the haplotype sequences. In contrast, the 2D layout of the HPRC PGGB chromosome 19 pangenome graph (Heumos *et al.*, 2024) clearly presents SVs, especially in the centromere’s location. Investigating these complex regions of a human chromosome is only possible when using long-read assemblies for graph construction.

5.7 *E.coli* data set

The 2146 full length E. coli sequences originate from Genbank (Sayers *et al.*, 2021) and were downloaded 18 months ago. The initial pangenome consisted of 2 graphical components (Suppl. Fig. 5.7). This means that no strong homologies were found in some sequences. There can be many reasons for additional graph components: (a) The chosen sequence identity during the WFMASH mapping was not low enough. Although we went for a low 90% sequence identity (as was done by Garrison *et al.* (2023)), we still observe this additional graph component, so its sequence must be quite dissimilar to all other sequences. (b) There is human contamination in the bacterial sequences (Breitwieser *et al.*, 2019). (c) Some sequences from GenBank may be of a low quality or were misassembled. We then used odgi explode to extract the largest graphical component, applied *odgi crush* and dropped all paths containing “plasmid” in their path name with *odgi paths*. This left us with one component and 2013 paths. In the 2D visualization, we observe a highly connected graph (Suppl. Fig. 5.7). All the reasons mentioned above, but especially horizontal gene transfer could explain this phenomenon. Therefore, there are a lot of edge crossings in the pangenome graph. The long stretches is dangling sequence. We speculate that here the 88 thousand Ns could play role.



Sample Name	Length	Nodes	Edges	Paths	Components ▾	A	C	T	G	N
ecoli_2146	140 465 404	6 118 071	8 972 252	2 146	2	21 671 240	19 111 201	21 699 33	19 131 791	58 851 837
ecoli2146.pan.explode	140 321 750	6 113 871	8 966 644	2 143	1	21 631 635	19 077 542	21 663 73	19 096 998	58 851 837
ecoli2146.pan.explode.crush	81 557 898	6 113 871	8 966 644	2 143	1	21 631 635	19 077 542	21 663 73	19 096 998	87 985
ecoli2146.pan.explode.crush.no_plasmids	81 557 898	6 113 871	8 966 644	2 013	1	21 631 635	19 077 542	21 663 73	19 096 998	87 985

Fig. S4. Excerpt of the 2146 sequences E. coli pangenome graph's MultiQC report. Displayed are vital graph statistics by MultiQC's ODGI module. A: Number of adenine bases in the graph. C: Number of cytosine bases in the graph. T: Number of thymine bases in the graph. G: Number of guanine bases in the graph. N: Number of bases with unknown base identity.

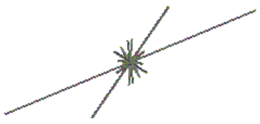


Fig. S5. odgi draw 2D layout visualization of the 2013 haplotypes E. coli pangenome graph.