ODGI: scalable tools for pangenome graphs

Andrea Guarracino, Dept. of Biology, University of Tor Vergata, Rome, Italy;
Simon Heumos, QBiC, University of Tübingen, Tübingen, Germany;
Pjotr Prins, Dept. Genetics Genomics & Informatics, UTHSC, Memphis, USA;
Erik Garrison, Dept. Genetics Genomics & Informatics, UTHSC, Memphis, USA

Abstract

Motivation: Pangenomes graphs address some of the shortcomings of mainline genomics, particularly reference genome bias and identification of complex structural variants. Hence, fast and versatile software is required to ask intricate questions of such data in an efficient way.

Results: Here we present ODGI, a novel range of tools that implements scalable algorithms and has an efficient in-memory representation of DNA graphs. ODGI includes tools for detecting complex regions, extracting *loci*, removing artifacts, exploratory analysis, manipulation, and visualisation. Its fast parallel execution facilitates routine pangenomic tasks as well as pipelines that can quickly answer complex biological questions of gigabase-scale pangenome graphs.

Availability: ODGI is C++ software published under the MIT license. Source code can be downloaded from https://github.com/pangenome/odgi, while the documentation is available at odgi.readthedocs.io. GUIX and Bioconda packages are provided, too.

Introduction

Thanks to advances in sequencing technology, new full-length genome assemblies are produced at a high rate, including efforts from the Human Pangenome Reference Consortium (HPRC) and telomere-to-telomere projects [1, 2, 3]. A pangenome can model a full set of genomic elements in a given species or clade. Pangenomics contrasts with reference genome-based approaches which relate sequences to a single consensus model [4]. A pangenome can be represented by a graph data structure incorporating sequences as nodes and their connections as edges. These nodes are shared for identical sequences, such as homologs, paralogs, and orthologs. So-called paths can represent reads, contigs, haplotypes, or individual chromosomes. A bi-directed graph can even contain both strands of DNA as paths. A

graph-based approach is particularly relevant for exploring highly repetitive regions and complex *loci*, such as the Major Histocompatibility Complex (MHC) *locus* (Fig. 1).

A major challenge is writing software that can deal with the sheer size and complexity of graphs representing hundreds of human genomes. The VG toolkit pioneered graph processing [5], but we found it does not necessarily scale for large data (see Table 1). We wrote a new toolset in C++ named Optimized Dynamic Genome/Graph Implementation (ODGI). ODGI implements an efficient graph structure in computer memory that can be dynamically updated using multiple CPU cores in parallel.

Table 1. Performance evaluation of ODGI and VG [5] using equivalent commands. The centromere region was extracted from CHR 6 of the HPRC year one assembly[1], i.e., 88 haploid, phased human genome assemblies from 44 individuals plus the chm13 cell line and GRCh38 reference genomes stored in a 4.3GB GFAv1 file. ODGI's path implementation can process paths in parallel, outperforming VG.

Operation	Command	Runtime (mm:ss)	Memory (GB)	File size (GB)
Convert GFA to native format				
	odgi build	1:35	10.39	5.4
	vg convert	8:14	28.43	6.1
Extract subgraph				
	odgi extract	1:15	9.43	2.7
	vg chunk	21:09	59.33	1.7

Results

ODGI can handle pangenome graphs in the Graphical Fragment Assembly (GFAv1[2]) format. It is designed to build and modify paths in parallel. Most of the tools are implemented to be applied together, piping the output from one tool into the next. Currently, ODGI comes with more than 30 tools, and here we present its key set:

- **odgi viz & draw**: Pangenome visualisation provides convenient insight into genomic variation. *odgi viz* generates a linearized representation of the pangenome (see Fig. 1a) and is capable of handling full length human chromosomes. *odgi draw* extends the visualization in 2D.

[1] https://github.com/human-pangenomics/HPP_Year1_Assemblies
[2] https://github.com/GFA-spec/GFA-spec/blob/master/GFA1.md

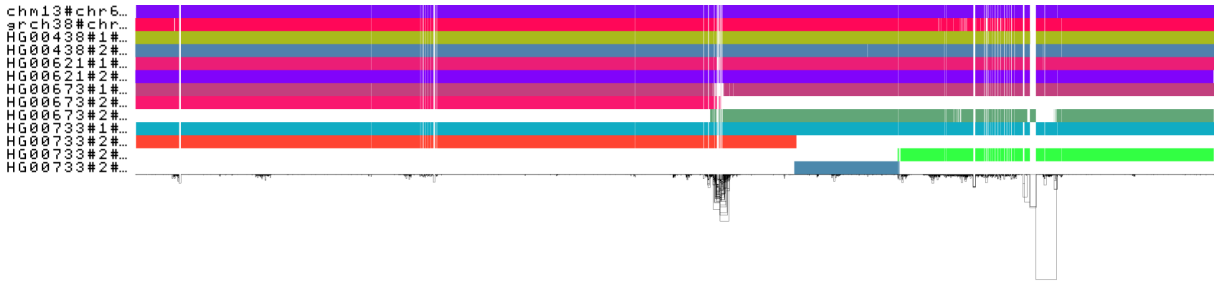
- **odgi stats, depth & degree**: Graphs statistics provide alternative ways to gain insight into pangenomes complexity. *odgi stats* returns the number of nodes, edges, paths, and graph length. *odgi depth* and *odgi degree* compute node depth and degree as defined by user-provided criteria. These methods allow the detection of complex regions generated by highly repetitive sequences.
- **odgi explode, squeeze & extract**: Pangenomes are constructed as large graphs. *odgi explode* separates units, such as chromosomes, into different files. *odgi squeeze* merges multiple graphs into the same file whilst preventing node ID collisions. *odgi extract* extracts regions of the graph as defined by certain criteria, allowing downstream processing of smaller subgraphs.
- **odgi position**: Pangenome graphs are flexible when it comes to coordinate systems. *odgi position* can use the coordinate system from a contained reference genome - a dynamic liftover - to display coordinates and other localised features (see Fig. 1b). Note that multiple reference genomes can be contained in the graph and any contained path can be used as a reference.
- **odgi untangle**: Alignment ambiguity in repetitive regions produces cycles in the pangenome graphs. *odgi untangle* produces a linearized overview of any collapsed locus by projecting paths into reference-relative BEDPE format output. This allows decomposing paralogy relationships.

Discussion

Pangenome graphs stand to become a ubiquitous tool in genomics [4]. With ODGI we implemented a state-of-the-art tool suite that can transform, analyse and visualise pangenome graphs at large scale. Lifting over annotations and linearizing nested graph structures place the suite as the bridge between traditional linear reference genome analysis and pangenome graphs. ODGI is a unique set of tools that enables scientists to explore and discover the underlying biology of pangenome graphs. Already, the tools are the backbone of pipelines such as the Pangenome Graph Builder (PGGB[3]) or nf-core/pangenome[4]. Future work will add support for RNA and protein sequences and expand on metadata capabilities of large pangenome graphs.

[3] https://github.com/pangenome/pggb
[4] https://github.com/nf-core/pangenome

a)



b)

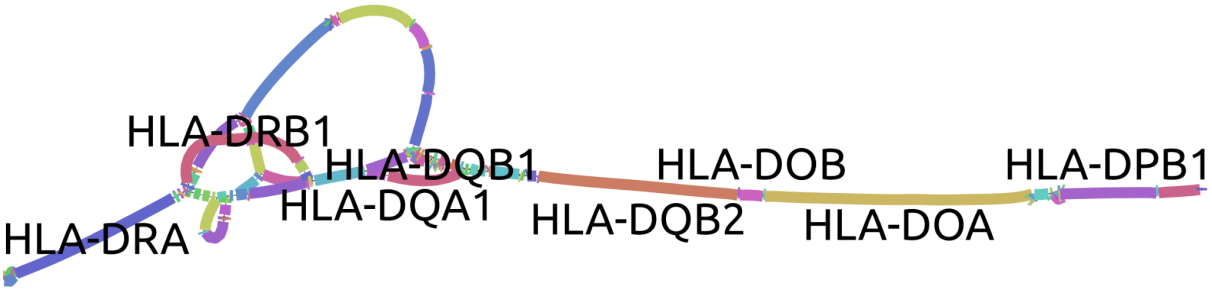


Fig 1. Pangenome visualisation of the MHC locus. (a) Projection of full MHC locus of 4 haploid phased human genome assemblies, plus the chm13 cell line and GRCh38 reference genome from the HPRC and created using *odgi viz*. The coloured bars represent the linearised paths. The black lines at the bottom represent the graph topology. (b) Graph representation of the same assemblies showing variations larger than 100 base pairs. Loops display contigs uniquely diverging from the consensus, caused by variation in paralogs/repeats. The gene labels are superimposed exploiting *odgi position* from a GRCh38 BED file and visualised by Bandage [6].

References

[1] Miga et al. “Telomere-to-telomere assembly of a complete human X chromosome”. *Nature* 585, 79–84 (2020). https://doi.org/10.1038/s41586-020-2547-7

[2] Logsdon et al. “The structure, function and evolution of a complete human chromosome 8”. *Nature* 593, 101–107 (2021). https://doi.org/10.1038/s41586-021-03420-7

[3] Nurk et al. “The complete sequence of a human genome”. *bioRxiv*, 2021.05.26.445798; doi: https://doi.org/10.1101/2021.05.26.445798

[4] Eizenga et al. “Pangenome Graphs”. *Annual Review of Genomics and Human Genetics*, 2020 21:1, 139-162

[5] Garrison et al. “Variation graph toolkit improves read mapping by representing genetic variation in the reference”. *Nature biotechnology*, vol. 36,9 (2018): 875-879. doi:10.1038/nbt.4227

[6] Wick et al. “Bandage: interactive visualisation of *de novo* genome assemblies”. *Bioinformatics*, 31(20) (2015): 3350-3352.