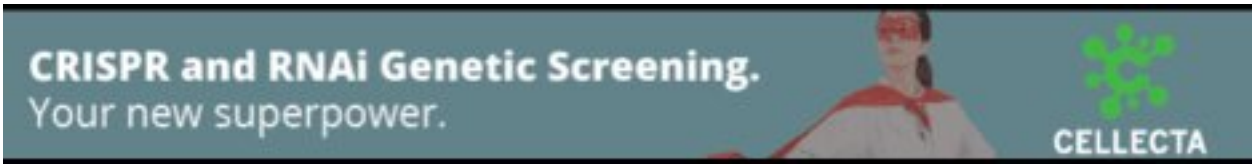
Interactive visualization and interpretation of pangenome graphs by linear reference –based coordinate projection and annotation integration

Zepu Miao and Jia-Xing Yue

Genome Res. 2025 35: 296-310 originally published online January 13, 2025
Access the most recent version at doi:10.1101/gr.279461.124



Method

Interactive visualization and interpretation of pangenome graphs by linear reference–based coordinate projection and annotation integration

Zepu Miao and Jia-Xing Yue

State Key Laboratory of Oncology in South China, Guangdong Key Laboratory of Nasopharyngeal Carcinoma Diagnosis and Therapy, Guangdong Provincial Clinical Research Center for Cancer, Sun Yat-sen University Cancer Center, Guangzhou 510060, China

With the increasing availability of high-quality genome assemblies, pangenome graphs emerged as a new paradigm in the genomic field for identifying, encoding, and presenting genomic variation at both the population and species level. However, it remains challenging to truly dissect and interpret pangenome graphs via biologically informative visualization. To facilitate better exploration and understanding of pangenome graphs toward novel biological insights, here we present a web-based interactive visualization and interpretation framework for linear reference–projected pangenome graphs (VRPG). VRPG provides efficient and intuitive support for exploring and annotating pangenome graphs along a linear-genome-based coordinate system (e.g., that of a primary linear reference genome). Moreover, VRPG offers many unique features such as in-graph path highlighting for graph-constituent input assemblies, copy number characterization for graph-embedding nodes, and graph-based mapping for query sequences, all of which are highly valuable for researchers working with pangenome graphs. Additionally, VRPG enables side-by-side visualization between the graph-based pangenome representation and the conventional primary linear reference genome–based feature annotations, therefore seamlessly bridging the graph and linear genomic contexts. To further demonstrate its functionality and scalability, we applied VRPG to the cutting-edge yeast and human reference pangenome graphs derived from hundreds of high-quality genome assemblies via a dedicated web portal and examined their local genome diversity in the graph contexts.

[Supplemental material is available for this article.]

Long-read sequencing technology has become the go-to-choice for most genome sequencing projects in recent years, empowering the production of chromosome-level telomere-to-telomere (T2T) genome assemblies for diverse organisms, including humans (Yue et al. 2017; Jiao and Schneeberger 2020; Nurk et al. 2022). With such a T2T reference assembly panel continuously expanding at both the population and species level, researchers began to use pangenome graphs to better represent the population- and species-wide genomic variation landscapes in a sequence-resolved manner (Eizenga et al. 2020). Compared with the conventional linear reference genome, pangenome graphs offer enhanced power and accuracy in read mapping and variant identification, especially in the presence of sequence polymorphisms and structural variants (SVs) (Paten et al. 2017). Therefore, a species-representative pangenome graph is expected to shed novel insights into the interpretation of the genotype-to-phenotype association and the discovery of missing heritability.

Although a number of tools have been developed to build pangenome graphs based on genome alignments, Minigraph (Li et al. 2020), Minigraph-Cactus (Hickey et al. 2024), and PGGB (Garrison et al. 2024) are among the most popular ones. Minigraph is designed for efficiently constructing a primary linear reference–based pangenome graph with large variants (e.g., SVs) compactly encoded in the reference graphical fragment assembly (rGFA) format. The rGFA format records the source information of each graph-embedding node relative to the input linear genomes and allows for traversing the pangenome graph along a stable coordinate system. This unique feature makes the pangenome graph built by Minigraph a natural and intuitive extension to the conventional linear reference genome, although a certain level of bias could be introduced during the graph building process regarding the choices of the primary linear reference and the input genome order (Garrison and Guarracino 2023). Minigraph preferentially considers large variants (e.g., SVs) while being less discriminative for small variants such as single-nucleotide variants (SNVs) and small insertion/deletions (indels) during graph construction; therefore, a pangenome graph constructed by Minigraph is not a strictly lossless representation of the full genomic variation carried by input genomes at the per-base level. As an improvement, Minigraph-Cactus was proposed as an extended solution that combines the compactness and efficiency of Minigraph as well as the base-level sensitivity and accuracy of Cactus. Pangenome graphs constructed by Minigraph-Cactus can effectively represent both small and large genomic variants while still preserving a trackable coordinate system derived from the primary linear reference genome. Finally, PGGB adopted an alternative approach for pangenome graph construction based on all-against-all genome alignments, which is theoretically unbiased but computationally more intensive. Also, PGGB will collapse the highly similar genomic segments of the input genomes into loops and therefore make the graph structure a bit less straightforward in referring back to the original linear input genomes.

Corresponding author: yuejiaxing@gmail.com

Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.279461.124. Freely available online through the *Genome Research* Open Access option.

35:296–310 Published by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/25; www.genome.org

An intuitive visualization of pangenome graphs can greatly assist the exploration and understanding of the global and local genomic variation in their graph representations. To date, several tools have been developed for visualizing pangenome graphs. Among them, Bandage (Wick et al. 2015), GfaViz (Gonnella et al. 2019), and gfaestus (https://github.com/chfi/gfaestus) focus more on large-scale graph topology, whereas SequenceTubeMap (Beyer et al. 2019), MoMI-G (Yokoyama et al. 2019), and PGR-TK (Chin et al. 2023) are more suitable for picturing fine-scale sequence variation. In addition, ODGI (Guarracino et al. 2022) showed improved performance on large-scale pangenome graphs with extended visualization function for binned and linearized one-dimensional local graph structure rendering. Another tool, PanGraphViewer (Yuan et al. 2023), while still being built for graph topology visualization, enables a coordinate-based graph querying and can be used for genomic variant examination when a VCF file is provided. Although these tools are quite useful in specific use scenarios, they almost exclusively focus on graphs per se, making it still challenging for general researchers to associate a pangenome graph with conventional linear genome assemblies and their feature annotations. Moreover, many of these existing tools fall short in scalability, lacking the capability of dynamically visualizing full-scale pangenome graphs on the fly. Therefore, a novel pangenome graph visualization framework is needed to address these shortcomings and to better bridge the linear-based and graph-based genomic representations, which will make pangenome graphs more accessible to a broader community.

Results

General design

In this study, we developed a web-based interactive visualization and interpretation framework for linear reference–projected pangenome graphs (VRPGs) with enhanced functionality and scalability. Functionality-wise, VRPG can present pangenome graphs along a stable linear coordinate system (e.g., that of a primary linear reference genome), therefore enabling browsing, querying, labeling, and highlighting pangenome graphs in a highly intuitive manner. For doing so, VRPG natively supports the rGFA-formatted pangenome graphs built by Minigraph while also shipping an auxiliary command-line module (gfa2view) to provide extended compatibility to pangenome graphs in GFAv1 format (e.g., those built from Minigraph-Cactus and PGGB). In addition to visualizing the pangenome graph itself, VRPG allows for user-defined annotation tracks alongside, which unifies the pangenome graph with various annotation data types under the same primary linear reference–based coordinate system. Regarding scalability, VRPG is algorithmically optimized to be capable of rapidly navigating, querying, and rendering large-scale pangenome graphs built upon hundreds of input genome assemblies. Multiple layout simplification options are further implemented to make it amenable for pangenome graphs with high complexity. Taken together, VRPG offers a novel and powerful way of visualizing pangenome graphs with unique strength in both functionality and scalability.

User interface and feature highlights

VRPG mainly operates via a web browser, in which users can easily navigate and interact with the rendered pangenome graph (Fig. 1). The user interface of VRPG consists of six panels: (1) the control panel, (2) the graph panel, (3) the genome coordinates and gene annotation panel, (4) the additional feature annotation panel, (5) the node information panel, and (6) the path information panel. Together, these panels offer a unified and engaging user experience for exploring pangenome graphs in a highly interactive and informative manner.

In the control panel, users can specify their interested pangenome graphs and query regions based on the predefined primary linear reference genome coordinates. VRPG can rapidly extract the corresponding subgraph accordingly and render it in five optional layouts (i.e., "ultra expanded" [default], "expanded," "squeezed," "hierarchical expanded," "hierarchical squeezed"), three graph simplification strategies (i.e., "nonref nodes" [default], "all nodes," and "none"), and an additional simplification parameter (i.e., "min bubble size," which is approximately the minimal cumulative node length within a bubble). These simplification options can be very useful when visualizing pangenome graphs built by Minigraph-Cactus or PGGB. In such pangenome graphs, base-level small variants are encoded as individual nodes, therefore resulting in a topology that is too complex to display unless properly simplified. Although the layout and simplification options described above already offer considerable flexibility for users to find the best rendering results, users can also manually drag any node in the displayed graph for layout fine-tuning when needed. Additional functions such as assembly-to-graph path highlighting (the "path highlighting" button) and sequence-to-graph mapping (the "sequence-to-graph mapping" button) can be further used for more advanced graph exploration.

In the graph panel, graph nodes corresponding to the predefined primary linear reference genome are always scaled according to their actual size and are typically shown along the center line, whereas nodes that represent genomic variants relative to the primary linear reference are shown alongside, which can be optionally scaled. Both the displayed nodes and edges are clickable. For nodes, the selected node will be thickened upon clicking, and its corresponding information will be reported in the node information panel. As for edges, the selected edge will be colored in red on the first click, and its color will revert to black on the second click. When the assembly-to-graph path highlighting feature is enabled, the traversing paths of the selected query assembly(-ies) will be marked in the graph, with all matched nodes and edges highlighted in red. The arrows on the matched edges reflect the traversing direction of the highlighted linear genome assemblies in the pangenome graph.

In the genome coordinates and gene annotation panel, a primary linear reference–based coordinate track is shown to provide positional reference to the pangenome graph rendered above. Together with this coordinate track, gene and mRNA annotation tracks are further depicted to provide more functional contexts. Users can find out the name of the displayed genes and mRNAs by mouse clicking (for genes) or hovering (for mRNAs).

In the additional feature annotation panel, users can specify one or more primary linear reference–based annotation features and visualize them together with the pangenome graph in a side-by-side and highly synchronized manner. Both qualitative and quantitative annotation features are supported here in the BED format. In this way, users can essentially display any annotation features (e.g., GC content, centromere, telomere, conserved elements, regulatory elements, genomic variants) and conveniently explore their physical association with the topology of pangenome graphs.

In the node information panel, a detailed summary report on node-associated information will be automatically generated upon

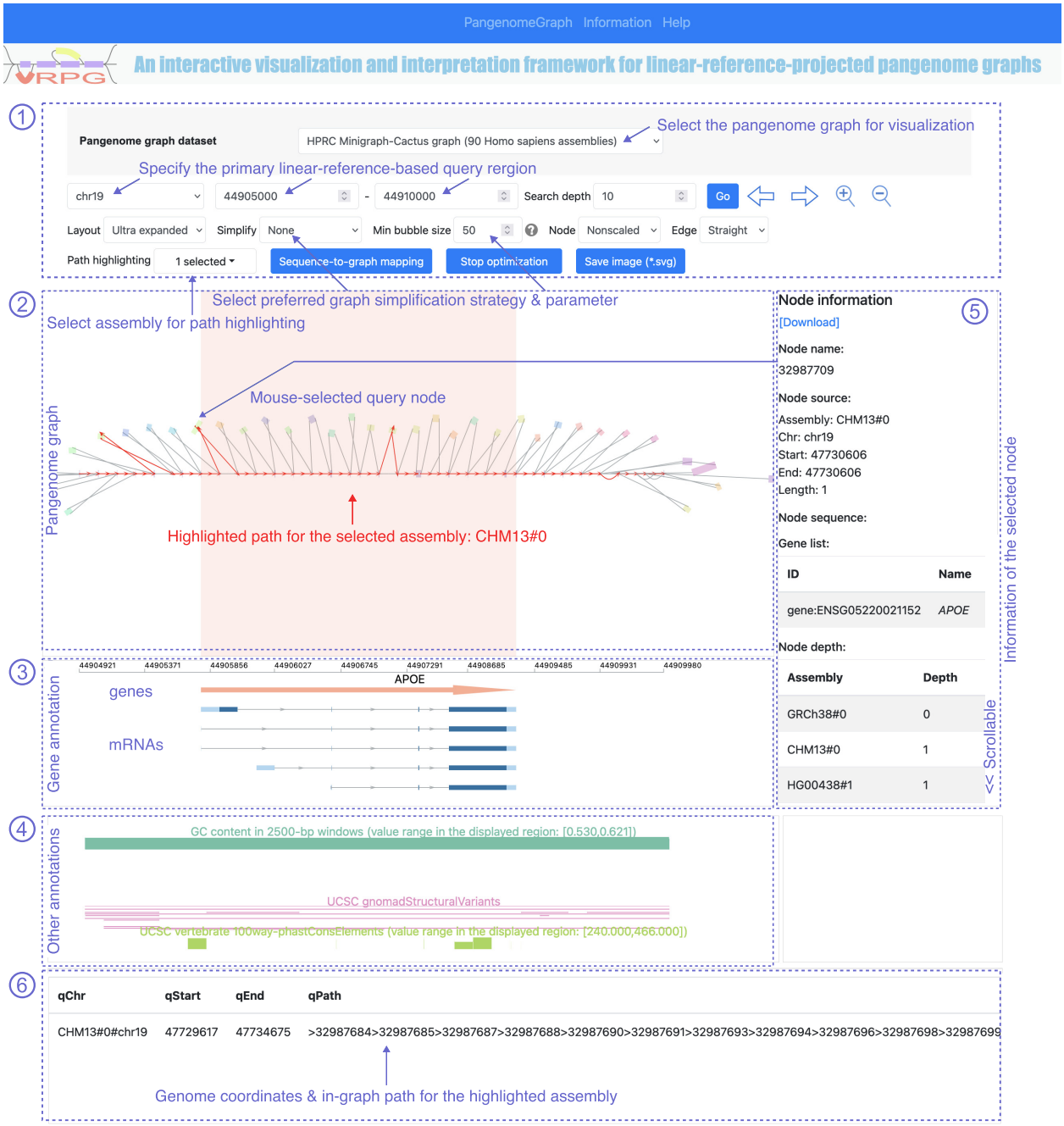


Figure 1. The interactive user interface of VRPG when opened in a web browser. After selecting the input pangenome graph data set, users can interactively visualize, navigate, and query the pangenome graph via a primary linear reference–based coordinate system. An example of the human *APOE* gene region (Chr 19: 44,905,000–44,910,000, GRCh38 coordinates) of the HRPC Minigraph-Cactus pangenome graph is shown here. The *APOE* gene bears strong associations with cardiovascular diseases and Alzheimer's disease. No layout simplification or node scaling was applied in VRPG for this visualization. Path highlighting for CHM13 was selected. All other options were left with defaults.

users' node selection in the graph. The reported information includes the assembly origin of the selected node, genes covered by this node, and estimated copy numbers of the corresponding genomic sequence across different graph-constituent assemblies.

In the path information panel, when users perform assembly-to-graph path highlighting (if only one assembly is selected) or sequence-to-graph mapping (currently only support pangenome graphs built by Minigraph), the returned results will be shown here describing the matched query genome coordinates, graph traversing path (denoted by directionally chained node IDs), and concise idiosyncratic gapped alignment report (CIGAR) string. This feature offers the precise measurement on how well the query sequence (specified for highlighting or mapping) matched with the pangenome graph, which is of great value for graph-based comparative genomic analysis but remains unavailable in any other pangenome graph visualization tools developed so far.

Genomic variant representations in VRPG

One major motivation of developing VRPG is to provide an intuitive way of associating the conventional linear assembly–based genomic variants to their graph representations. Now it comes in handy with the assembly-to-graph path highlighting feature of VRPG. In Figure 2, we exemplified how different types of genomic variants from the query genome relative to the primary linear reference are typically depicted by VRPG in the context of a pangenome graph: divergent region (Fig. 2A), insertion (Fig. 2B), deletion (Fig. 2C), duplication (Fig. 2E), and duplication with divergent region (Fig. 2F). Because VRPG typically shows nodes representing the primary linear reference assembly along the center line, any highlighted query genome path that departs from this

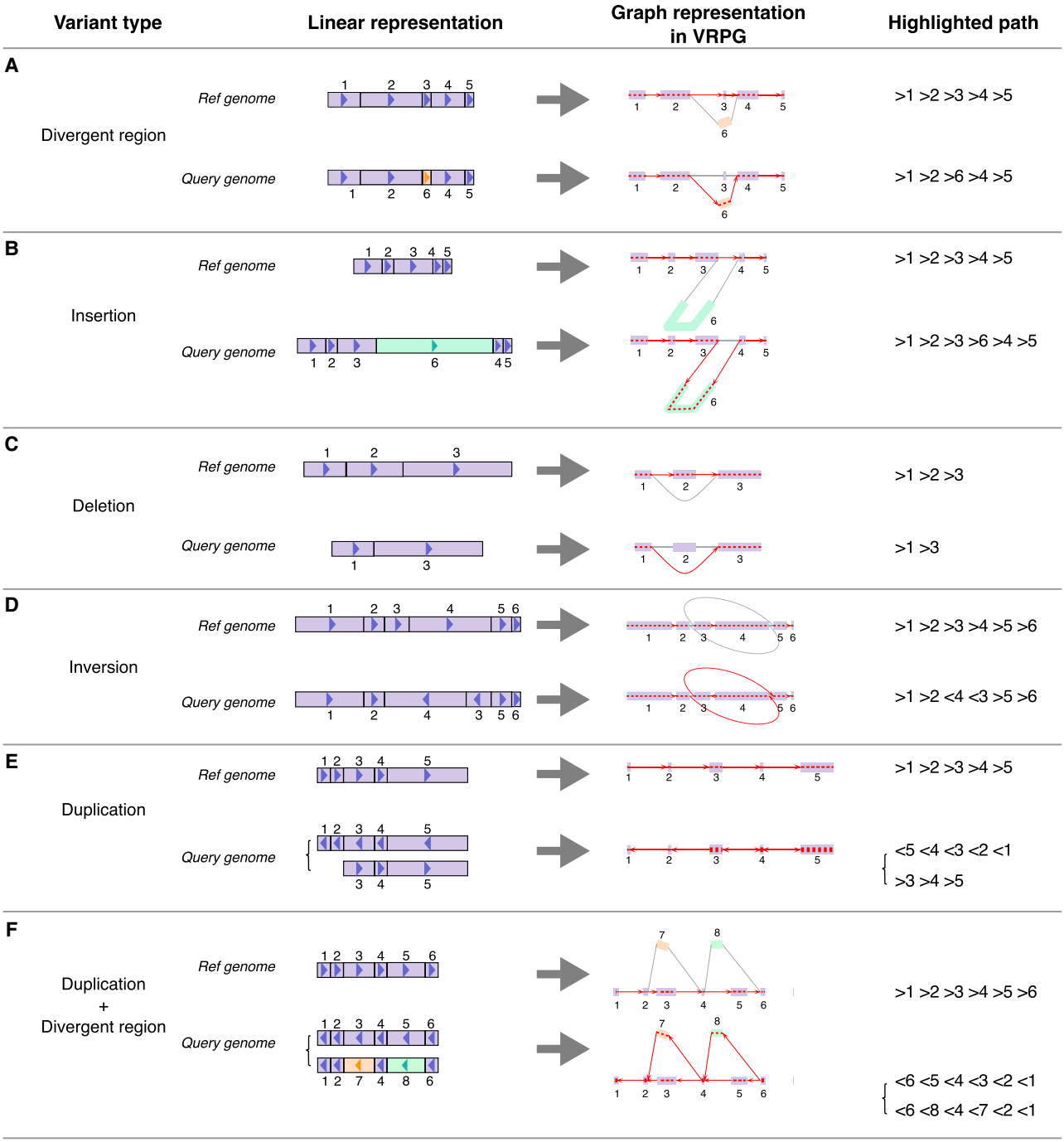


Figure 2. VRPG's visualization of different types of genomic variants. For different types of genomic variants such as divergent region (*A*), insertion (*B*), deletion (*C*), inversion (*D*), duplication (*E*), and duplication with divergent region (*F*), their VRPG-based graph representations are depicted with a path highlighting feature enabled for the reference and query genome, respectively. Information of the highlighted path regarding nodes' traversing order and orientation is further reported (in GAF-specified style) in VRPG's path information panel.

center line implies the occurrence of genomic variant(s). Moreover, because VRPG's highlighting function also depicts the traversing direction of each path from one node to another, it is straightforward to determine the genomic orientation of the variant(s) observed (e.g., in the case of an inversion [Fig. 2D]). In addition, the in-node paths will also be thickened in a two-state manner (thin vs. thick) if the highlighted genome assembly traverses the corresponding node(s) more than once, which implies the occurrence of duplications (Fig. 2E,F). Aside from these graphical indications, the detailed traversing paths of the highlighted assembly are also reported in the path information panel of VRPG, which provides additional help for variant interpretation. The traversing path denotation used by VRPG follows the graph alignment format (GAF) specification.

Application demonstration 1: visualizing a pangenome graph derived from 163 yeast genome assemblies

To demonstrate the application of VRPG in real world examples, we set up a dedicated webserver (https://www.evomicslab.org/app/vrpg/) to visualize a reference pangenome graph derived from 163 budding yeast genome assemblies. Here, we employed Minigraph to construct this pangenome graph using the *Saccharomyces cerevisiae* reference assembly panel (ScRAP) that we recently assembled (O'Donnell et al. 2023; Miao et al. 2024). This yeast pangenome graph consists of 37,062 nodes and 52,756 edges, with a total length of 27,190,479 bp.

The flip/flop inversion

Based on this yeast pangenome graph, we used VRPG to visualize a famous flip/flop inversion region on the Chromosome XIV (Chr XIV) of the *S. cerevisiae* genome. This flip/flop inversion region is flanked by two 4.2 kb inverted repeat (IR) regions and remains polymorphic in *S. cerevisiae* and its sister species in the genus *Saccharomyces* (Salzberg et al. 2022). For example, within *S. cerevisiae*, our previous study revealed that the genome of the Sake strain Y12 shares conserved synteny with the *S. cerevisiae* reference genome (SGDref) within this region, whereas the genome of the North American strain YPS128 shows an inversion instead (Fig. 3A; Yue et al. 2017). In accordance with this prior knowledge, by enabling the assembly path highlight feature of VRPG, we recaptured such polymorphic inversion in the pangenome graph (Fig. 3B–E). As illustrated by VRPG, both SGDref and Y12 revealed a simple linear assembly-to-graph path through the nodes along the central line, suggesting their primary linear reference–like sequence structure in this region (Fig. 3B,D). In contrast, YPS128 shows an S-shaped path in the flip/flop region, which implies the existence of the inversion (Fig. 3E).

The DOG2 deletion

The yeast paralog gene pairs *DOG1* and *DOG2* encode 2-deoxyglucose-6-phosphate phosphatase involved in glucose metabolism. They are homologous to the human *PUDP* (previously known as *HDHD1*) gene, an anticancer treatment target for intervention in the glycolytic metabolism of tumor cells (Defenouillère et al. 2019). Previously, we have identified a polymorphic deletion for the *DOG2* gene in the comparison of seven representative yeast strains using their T2T genome assemblies (Yue et al. 2017). For example, this gene is present in strains like S288C and W303 but is absent in other strains such as SK1 and Y12 (Fig. 4A). Here, we used VRPG to visualize this gene presence/absence variation in the context of a pangenome graph (Fig. 4B–F). With the SGDref (S288C) as the primary linear reference, the path taken through the graph by the linear assembly W303 is simple, suggesting its reference-like sequence structure in this region. In contrast, the paths taken by SK1 and Y12 clearly depart from the SGDref path by bypassing the node representing the *DOG2* gene region.

Application demonstration 2: visualizing pangenome graphs derived from 90 human genome assemblies

As a further proof for VRPG's versatility and scalability, we retrieved the human pangenome graphs generated by the Human Pangenome Reference Consortium (HPRC) based on 90 human genome assemblies (Liao et al. 2023) and visualized them via our VRPG demonstration webserver (https://www.evomicslab.org/app/vrpg/). Starting with the same input genome set, HPRC built three human pangenome graphs with Minigraph, Minigraph-Cactus, and PGGB, respectively. The HPRC human pangenome graph built by Minigraph consists of 391,950 nodes and 566,204 edges, with a total length of 3,198,196,033 bp. In comparison, the total node and edge numbers of graphs built by Minigraph-Cactus (80,069,733 nodes and 110,938,345 edges) and PGGB (110,884,673 nodes and 154,756,169 edges) are substantially larger because small variants such as SNV and indels are fully considered by Minigraph-Cactus and PGGB during their graph construction, whereas Minigraph predominantly considers larger variants such as SVs.

The DSCAM intronic inversion

For the demonstration, we used VRPG to visualize an inversion located between the exon 32 and exon 33 of the *DSCAM* gene (Fig. 5; Audano et al. 2019). This inversion is mediated by two inverted 6.0 kb L1 elements (Fig. 5A), forming a structure much like the yeast Chr XIV flip/flop inversion described above. *DSCAM* gene belongs to the immunoglobulin superfamily of cell adhesion molecules (Ig-CAMs) and functions in central and peripheral nervous system development. The overexpression of *DSCAM* promotes the development of Down syndrome (DS) and congenital heart disease (DSCHD) (Grossman et al. 2011; Liu et al. 2023). This *DSCAM* inversion is segregated between the human reference assembly GRCh38 and the T2T assembly CHM13 (Fig. 5B). Using VRPG's assembly path highlighting feature, we compared the paths of GRCh38 and CHM13 in HPRC human pangenome graphs built from Minigraph, Minigraph-Cactus, and PGGB (Fig. 5C–E). In all graphs, the GRCh38 shows a simple linear path as expected. The CHM13 T2T assembly, on the other hand, shows the characteristic S-shaped pattern in both the Minigraph and Minigraph-Cactus pangenome graphs for the inverted region. In the PGGB graph, the pattern is more complex, likely owing to the sequence collapsing procedure during PGGB's graph construction. Nevertheless, clear distinctions between the GRCh38 and CHM13 paths can still be found, as CHM13 shows a unique protruding loop in the inverted region. Moreover, given that the pangenome graphs built from Minigraph-Cactus and PGGB tend to be much more complex in topology, here we used this case to show the effectiveness of VRPG's layout simplification feature. When no layout simplification was applied, we can see signals of many genomic variants (indicated with small triangles in Fig. 6) in the Minigraph-Cactus and PGGB graphs. When we enabled the nonreference node simplification, most such signals disappeared as they represent small variants (i.e., <50 bp), leaving the large variants (e.g., our interested inversion) more noticeable even without zooming-in.

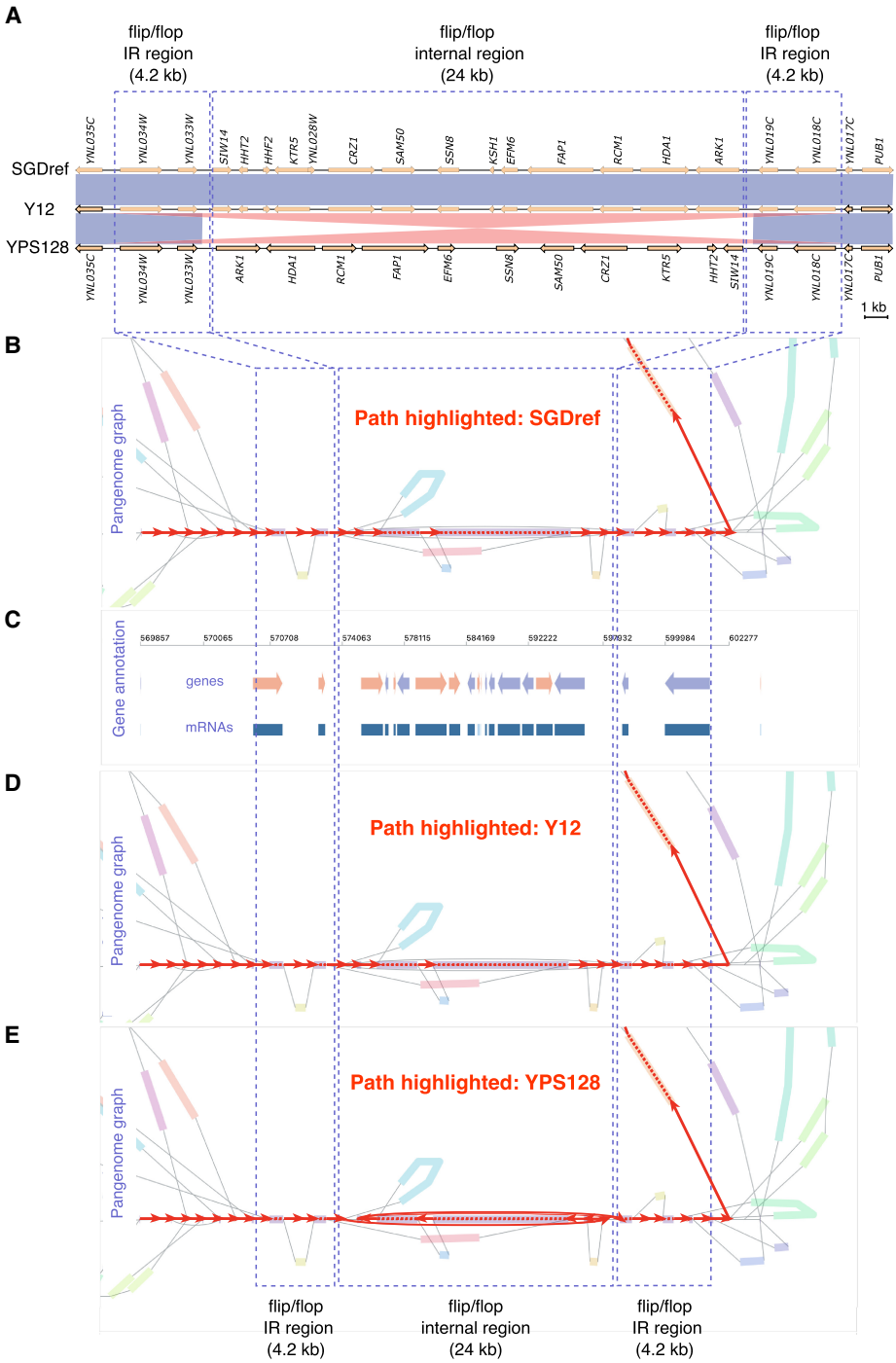


Figure 3. VRPG's visualization of the Chromosome XIV (Chr XIV) flip/flop region in a yeast pangenome graph. (*A*) The genome sequence and synteny comparison among SGDref (S288C), Y12, and YPS128 for the Chr XIV flip/flop region, with the red and blue shades representing homologous regions shared with >97% sequence similarity. Blue indicates same direction as shown in the flip/flop IR regions; red, reverse direction as shown in the flip/flop internal region. (*B–D*) VRPG visualization for the graph in this region (Chr XIV: 569,857–602,365, SGDref coordinates). The assembly paths of SGDref (*B*), Y12 (*D*), and YPS128 (*E*) are highlighted, respectively, with the SGDref-based coordinate system and gene annotation track (*C*) further shown in between. No layout simplification or node scaling was applied in VRPG for this visualization. All other options were left with defaults.

The CR1 intragenic deletion

The *CR1* gene (also known as *CD35*) is a complement activation receptor gene that is strongly associated with Alzheimer's disease (Lambert et al. 2009; Brouwers et al. 2012). This gene is structurally polymorphic in human populations, bearing an intragenic duplication region whose copy number correlates with the risk of Alzheimer's disease (Kucukkilic et al. 2018). An 18.6 kb intragenic deletion has been reported for *CR1* in a recent comparison between the human GRCh38 reference genome and the CHM13

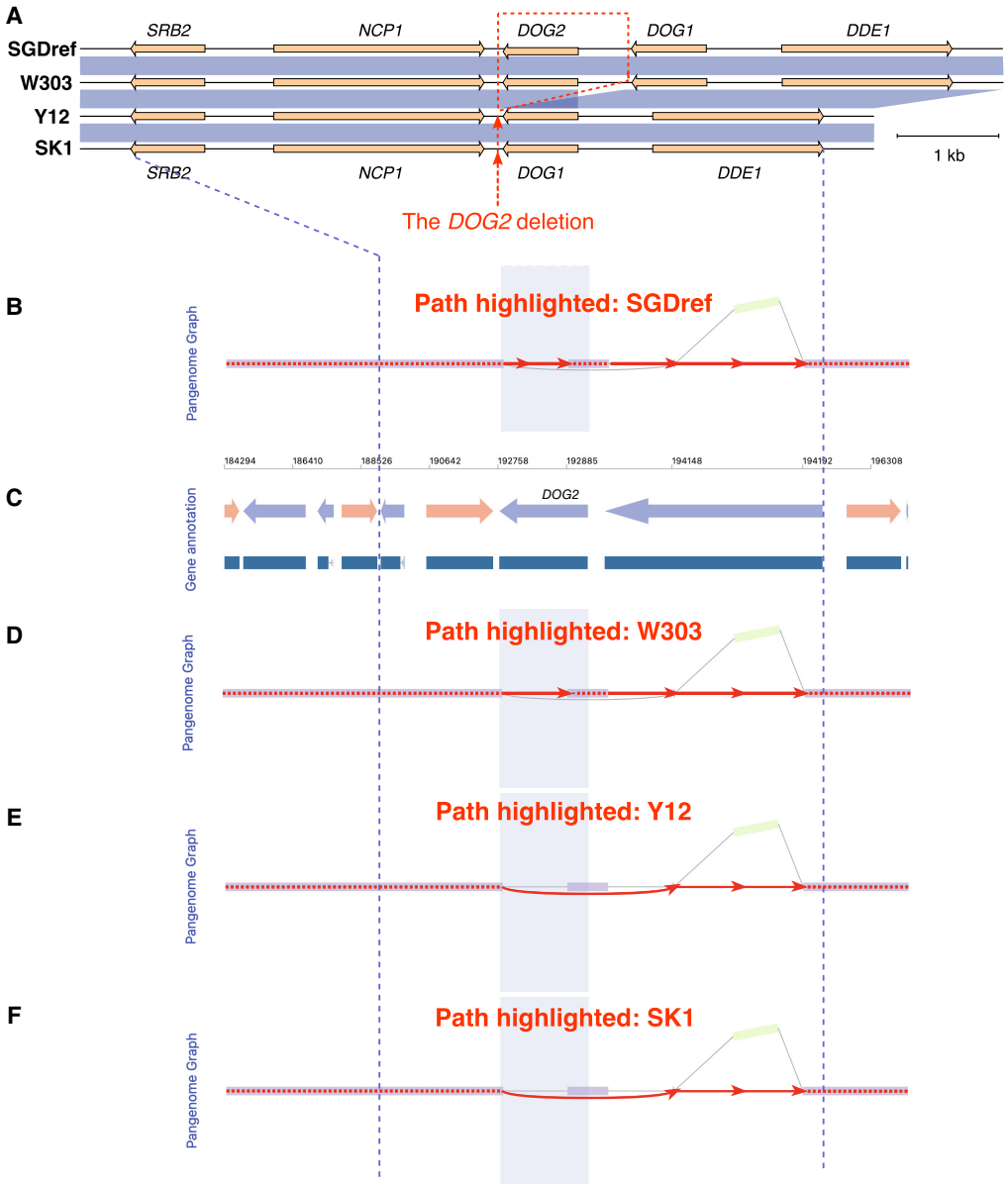


Figure 4. VRPG's visualization of the polymorphic *DOG2* deletion in a yeast pangenome graph. (*A*) The genome sequence and synteny comparison among SGDref, W303, SK1, and Y12 for the *DOG2* region, with the blue shades representing homologous regions shared with >97% sequence similarity. (*B–F*) VRPG visualization for the graph in this region (Chr VIII: 189,131–197,233, SGDref coordinates). The assembly paths of SGDref (*B*), W303 (*D*), Y12 (*E*), and SK1 (*F*) are highlighted, respectively, with the SGDref-based coordinate system and gene annotation track (*C*) further shown in between. The light purple shades denote the *DOG2* gene region. No layout simplification or node scaling was applied in VRPG for this visualization. All other options were left with defaults.

T2T genome, with the latter one hosting the shorter version (Yang et al. 2023). Here, we took a closer examination for this deletion in the context of its nearby sequence homology and confirmed that this deletion locates within the previously reported intragenic duplication region (Fig. 6A,B). Next, we used VRPG to visualize this *CR1* intragenic deletion based on the human pangenome graphs built by Minigraph, Minigraph-Cactus, and PGGB, respectively (Fig. 6C–E). By comparing the highlighted paths of GRCh38 and CHM13, this deletion can be clearly identified in both Minigraph and Minigraph-Cactus pangenome graphs. As for the PGGB graph, VRPG also depicted notable path differences between GRCh38 and CHM13 (note those extra red paths highlighted for CHM13), although the PGGB graph is topologically too complex to reflect the deletion in an intuitive way.

Functionality and performance comparison with other pangenome graph visualization tools

As briefly described in the introduction, there have been several tools developed for visualizing pangenome graphs. A representative list of these tools may include Bandage, GfaViz, gfaestus, SequenceTubeMap, MoMI-G, ODGI, PGR-TK, and PanGraph-Viewer. Here, we summarized their basic information and design features in a comparison table (Table 1). In comparison, VRPG

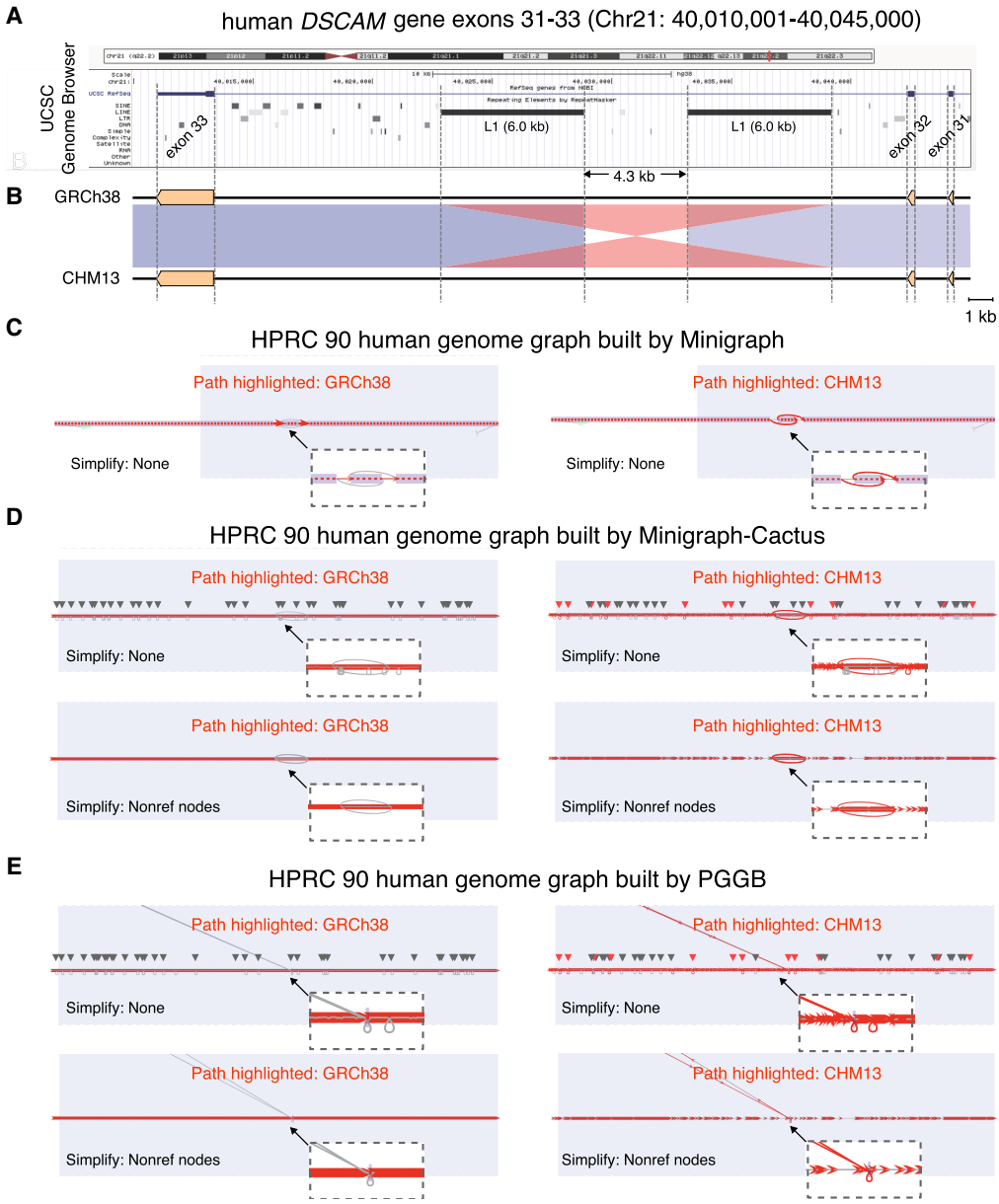


Figure 5. VRPG's visualization of the *DSCAM* intronic inversion in the HPRC human pangenome graphs. (*A*) The University of California, Santa Cruz (UCSC) Genome Browser view of the *DSCAM* exons 31–33 region at Chr 21: 40,010,001–40,045,000 (GRCh38 coordinates) with annotation tracks for chromosome ideogram, gene structure, and repetitive sequence features displayed. (*B*) The genome sequence and synteny comparison between GRCh38 and CHM13 for the *DSCAM* exons 31–33 region, with the blue (matched in same directions) and red (matched in reversed directions) shades representing homologous regions shared with >98% sequence similarity. (*C–E*) VRPG visualization for the *DSCAM* inversion in the HPRC human pangenome graphs derived from Minigraph (*C*), Minigraph-Cactus (*D*), and PGGB (*E*), with the assembly paths of GRCh38 (*left*) and CHM13 (*right*) highlighted, respectively. The light purple shades denote the *DSCAM* exons 31–33 region. The small triangles *along* the graph indicate the trace of genomic variants shown in the graph, with those corresponding to the highlighted path further colored in red. VRPG's squeezed layout was used when visualizing the Minigraph-Cactus and PGGB graphs. Layout simplification was applied as indicated, whereas node scaling was enabled in all cases. All other options were left with defaults.

shines in its all-around input compatibility, web-based dynamic display, and versatile functionalities. Especially, being able to carry out cross-examination between the pangenome graph and linear assemblies is critical for understanding and utilizing a pangenome graph, which is currently best supported in VRPG. Moreover, although several tools can take additional genome annotation files as the inputs, how they can actually utilize these files vary substantially. For example, Bandage can use such annotation files for region-specific graph highlighting, whereas PanGraphViewer uses them for subgraph extraction instead. VRPG is the only tool that enables a highly interactive and synchronized visualization between a pangenome graph and rich genomic feature annotations, helping to place the somewhat abstract pangenome graph into a more intuitive and biologically informative context.

Aside from the comparison in general designs and functionalities, we also benchmarked the computational performance of

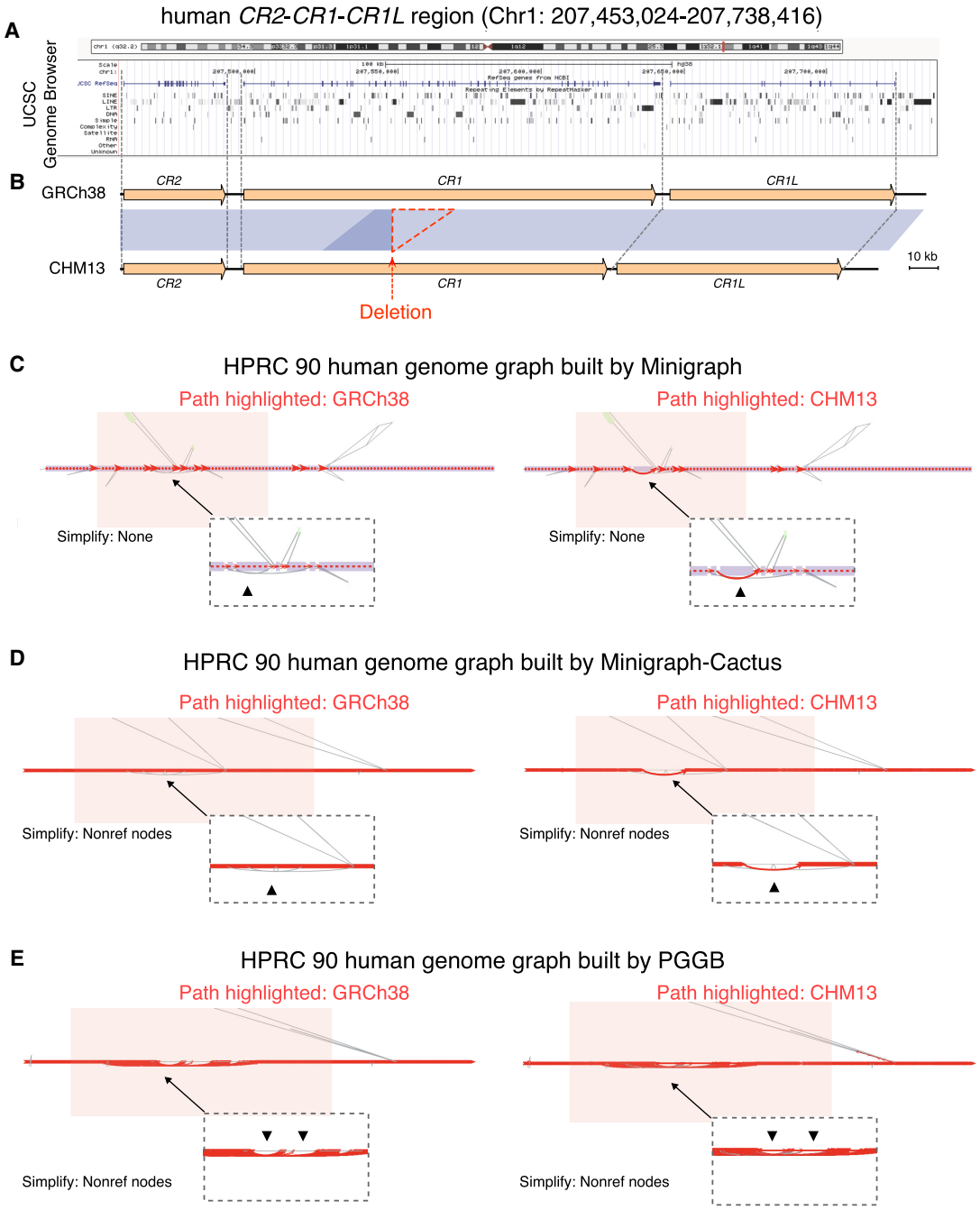


Figure 6. VRPG's visualization of the *CR1* intragenic deletion in the HPRC human pangenome graphs. (*A*) The UCSC Genome Browser view for the *CR2-CR1-CR1L* region at Chr 1: 207,453,024–207,738,416 (GRCh38 coordinates) with annotation tracks for chromosome ideogram, gene structure, and repetitive sequence features displayed. (*B*) The genome sequence and synteny comparison between GRCh38 and CHM13 for the *CR2-CR1-CR1L* region, with the blue shades representing homologous regions shared with >98% sequence similarity. The deletion is further highlighted in the red triangle. (*C–E*) VRPG's visualization for the *CR1* deletion in the HPRC human pangenome graphs derived from Minigraph (*C*), Minigraph-Cactus (*D*), and PGGB (*E*), with the genome paths of GRCh38 and CHM13 highlighted, respectively. The pink shades denote the *CR1* genic region. The path differences in the deletion region are further indicated by black triangles. VRPG's squeezed layout was used for the Minigraph-Cactus and PGGB graph visualization. Nonref node simplification was applied as indicated when visualizing the Minigraph-Cactus and PGGB graphs. All other options were left with defaults.

VRPG against Bandage, GfaViz, and PanGraphViewer for the yeast and human pangenome graph visualization (Table 2). All of these four tools support GFAv1/rGFA-formatted pangenome graphs, which makes them more comparable than other tools. First, we tested if these tools could parse (no need for rendering) the full-scale yeast and human pangenome graphs used in this study. The Chr 22 subset of the full-scale human graphs was also used for additional comparison. Note that by design, VRPG performs format transformation (when the input graph is in GFAv1 format) and indexing in advance, which are its heavy-lifting steps, and

Table 1. Basic information and functionality comparison between VRPG and other pangenome graph visualization tools

	VRPG	Bandage	GfaViz	gfaestus	Sequence TubeMap	MoMI-G	ODGI[a]	PGR-TK[a]	PanGraphViewer
Distributing license	MIT	GNU GPL v3	ISC	MIT	MIT	MIT	MIT	MIT	MIT
Supported operating system	Linux	Linux; MacOS; Windows	Linux; MacOS; Windows	Linux; MacOS	Linux; MacOS	Linux; MacOS	Linux	Linux; MacOS	Linux; MacOS; Windows
Supported pangenome graph format	GFAv1; rGFA	GFAv1; rGFA;	GFAv1; GFAv2; rGFA	GFAv1 (also needs ODGI's layout TSV)	xg; vg; gbz	xg	GFAv1; og	GFAv1	GFAv1; rGFA
Deployment	Web-based	Desktop-based	Desktop-based	Desktop-based	Web-based	Web-based	Desktop-based	Desktop-based	Web-based; Desktop-based
Rendering object	Subgraph	Full graph	Full graph	Full graph	Subgraph	Subgraph	Subgraph	Subgraph	Subgraph
Graph query mode	By coordinates	By node IDs	By node IDs	By node IDs	By coordinates; by node IDs	By coordinates	By coordinates	N/A	By coordinates
Rendering style	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Static	Static	Dynamic
Support for annotation[b]	BED; GFF3	TSV	N/A	BED; TSV	BED	BED; GFF3; WIG	N/A	N/A	BED; GFF3; GTF
Sequence-to-graph mapping	Yes (for graphs built by Minigraph)	Yes	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Assembly-to-graph path highlighting	Yes	N/A	N/A	N/A	N/A	N/A	Yes (partial)	N/A	N/A
Assembly-specific node depth report	Yes	N/A	N/A	N/A	N/A	N/A	Yes (bin-based)	N/A	N/A

[a]Tools like ODGI and PGR-TK are multipurpose software suites, both of which perform graph visualization via their submodules. The function comparison list here only concerns their visualization module.
[b]Although multiple tools accept user-supplied annotation files, they vary considerably regarding how they utilize these files with the graph.

consumes more computational resources. Considering that these steps only need to be executed once and they can be fully prepared before the actual graph parsing and visualization, these steps were not included in our benchmarking analyses (for an estimated resource consumption based on HPRC human pangenome graphs, see Methods). Mechanistically, VRPG, Bandage, and PanGraphViewer all adopted a two-step strategy for graph visualization, with the parsing and rendering steps fully decoupled. In contrast, GfaViz parses and renders the graph in a single step, thus consuming substantially more running time and memory. As a result, GfaViz was not able to parse the full-scale human graphs, nor their Chromosome 22 (Chr 22) subsets, as it quickly hit the memory cap of our testing machine (16 GB). Bandage also struggled with the full-scale Minigraph-Cactus and PGGB graphs owing to memory limitation. PanGraphViewer has better memory management but appears less robust when parsing graphs built by Minigraph-Cactus and PGGB. In comparison, VRPG showed a highly stable and robust performance when parsing all tested pangenome graphs once the corresponding graphs had been properly converted and indexed in advance.

Next, we compared the computational resource consumption of graph rendering for specific genomic regions. Four regions were selected for this test: two from the yeast graph and two from the human graphs. As for the full-graph, we found GfaViz is more sensitive to graph complexity as it failed to render the Minigraph-Cactus-based and PGGB-based subgraph for the 500 kb testing region (Chr 22: 20,000,001–20,500,000) in humans. Also, PanGraphViewer seems lacking the support for visualizing the PGGB-based subgraph as observed in our full-graph test. Both Bandage and VRPG showed robust performance in subgraph rendering, with VRPG further shining in rendering speed, thanks to its block index design (see Methods).

Discussion

As genome sequencing technologies keep progressing toward longer read length and higher per-base accuracy, generating reference-quality genomes at a population scale is becoming increasingly affordable (De Coster et al. 2021). Along with this trend, researchers began to develop new computational frameworks to accommodate and analyze such growing collections of high-quality genomes with better efficiency. The pangenome graph serves as a compact yet extensible data structure that describes population-level genetic diversity at both the base and structure levels via a graph representation (Eizenga et al. 2020). This is a highly active research field with new algorithms and applications rolling out very quickly.

Table 2. Performance comparison between VRPG and other pangenome graph visualization tools

Benchmarking task	Graph source	Graph type	Graph builder	Graph complexity (N:node; E:edge)	VRPG[a]	Bandage	GfaViz	PanGraphViewer
Graph parsing	Yeast	Full-graph	Minigraph	N: 37,062 E: 52,756	0.10 sec 64.80 MB	3.50 sec 254.11 MB	247.11 sec 2769.50 Mb	0.22 sec 282.10 MB
Graph parsing	Human	Full-graph	Minigraph	N: 424,643 E: 637,628	0.07 sec 54.44 MB	80.35 sec 10,066.50 MB	NA (memory cap)	7.83 sec 524.38 MB
			Minigraph-Cactus	N: 80,069,733 E: 110,938,345	0.10 sec 64.73 MB	NA (memory cap)	NA (memory cap)	NA (parsing error)
			PGGB	N: 110,884,673 E: 154,756,169	0.09 sec 58.26 MB	NA (memory cap)	NA (memory cap)	NA (parsing error)
Graph parsing	Human	Subgraph > Chr 22	Minigraph	N: 9545 E: 14,307	0.01 sec 62.5 MB	1.80 sec 260.15 MB	242.35 sec; 10,083.49 MB	0.19 sec 307.11 MB
			Minigraph-Cactus	N: 1,279,308 E: 1,780,703	0.01 sec 58.16 MB	84.50 sec 2,120.66 MB	NA (memory cap)	7.88 sec 815.58 MB
			PGGB	N: 1,124,300 E: 1,568,024	0.02 sec 56.11 MB	71.05 sec 1890.45 MB	NA (memory cap)	NA (parsing error)
Graph rendering	Yeast	Subgraph > Chr IV: 1–50,000	Minigraph	N: 103 E: 142	0.02 sec 82.96 MB	0.76 sec 105.94 MB	0.12 sec 171.48 MB	1.08 sec 323.73 MB
		Subgraph > Chr IV: 1–500,000	Minigraph	N: 1155 E: 1641	0.02 sec 194.86 MB	1.51 sec 122.36 MB	3.66 sec 324.79 MB	1.68 sec 324.70 MB
Graph rendering	Human	Subgraph > Chr 22: 20,000,001–20,100,000	Minigraph	N: 13 E: 18	0.01 sec 55.24 MB	0.30 sec 103.70 MB	0.16 sec 171.23 MB	0.01 sec 255.26 MB
			Minigraph-Cactus	N: 2248 E: 3106	0.05 sec 219.78 MB	1.81 sec 129.66 MB	13.25 sec 772.58 MB	1.88 sec 339.91 MB
			PGGB	N: 2252 E: 3116	0.05 sec 237.48 MB	1.68 sec 129.81 MB	14.15 sec 785.69 MB	NA (parsing error)
		Subgraph > Chr 22: 20,000,001–20,500,000	Minigraph	N: 288 E: 435	0.03 sec 82.78 MB	1.33 sec 111.81 MB	0.36 sec 181.08 MB	1.29 sec 321.28 MB
			Minigraph-Cactus	N: 21,982 E: 30,639	0.18 sec 605.82 MB	19.12 sec 347.27 MB	NA (memory cap)	7.37 sec 338.50 MB
			PGGB	N: 66,132 E: 94,595	0.33 sec 1270.61 MB	69.20 sec 766.10 MB	NA (memory cap)	NA (parsing error)

[a]The format conversion (for Minigraph-Cactus and PGGB graphs) step and index building step were performed in advance and were not included in this benchmarking.

One can envision that many genomic analyses that currently rely on conventional linear reference genomes will be eventually carried out based on pangenome graphs in future. A better understanding of how different layouts and topologies of pangenome graphs correspond to the actual population-wide genomic variation is of critical importance for correctly interpreting pangenome graph–based analysis results.

In this study, we developed VRPG, a web-based interactive framework for pangenome graph visualization and interpretation with full scalability, capable of rendering complex pangenome graphs derived from hundreds of genome assemblies. In addition to intuitive graphical visualization, VRPG also provides native supports for integrating conventional linear-genome-based feature annotations, enabling seamless examination of genomic variation in both graph-based and linear-based contexts. As further demonstrated with real world examples of yeast and human pangenome graphs as well as with benchmarking comparison with other tools, VRPG shines with its unique advantages in both functionality and scalability, making it a highly compelling choice for interactive and informative pangenome graph visualization.

Although efforts have been taken, there are also some limitations for VRPG in its current form. For example, some of its display layouts can be further improved for better clarity, especially when the graph topology becomes complex. Its sequence-to-graph mapping feature currently only works for the pangenome graphs built by Minigraph, whereas similar functionality is yet to be developed for graphs built by Minigraph-Cactus and PGGB. In the same vein, a broader graph compatibility beyond the rGFA and GFAv1 formats will be quite helpful for extending VRPG's application scope. Finally, the installation of VRPG has only been tested in a Linux-based environment (e.g., CentOS and Ubuntu) so far. Additional installation compatibility with MacOS should be possible but has yet to be explored. That said, given the web-based design of VRPG, users can access them remotely from any web-enabled platform via a centralized installation, which should help to alleviate this limitation. In the near future, with the help from the ever-growing pangenome graph community, we anticipate VRPG to be further improved, facilitating researchers from different fields to better explore the power of graph-based pangenomics, especially in the age of large-scale long-read-based population sequencing

(Rech et al. 2022; O'Donnell et al. 2023; Weller et al. 2023; Gustafson et al. 2024; Lian et al. 2024).

Methods

Hardware and software recommendations

VRPG is designed for a web-enabled desktop or computing server running the Linux operating system. Regarding the hardware, the resource-intensive steps are graph2view transformation (when the input graph is GFAv1-formatted) and index building. For example, for the HPRC Minigraph-Cactus graph, these two steps together took ~4 h with 10 CPU threads and 16 GB memory on a computing server equipped with Intel Xeon gold 6248R CPU (at 3.00 GHz). Therefore, we recommend executing these two steps on a computing server in advance when the input pangenome graph is large and complex. But once the format transformation and index building are processed, very few resources are further needed to access and render the indexed graph within VRPG's framework. Even with the HPRC human pangenome graphs, an ordinary desktop with 4 GB memory should be enough to render the subgraph for any given region that is ≤1 Mb. As for the software, VRPG has been tested with both CentOS and Ubuntu Linux operating systems. A list of third-party software dependencies for VRPG's installation, compilation, and execution is also provided (Supplemental Table S1).

Naming scheme of input assemblies

For VRPG, the naming scheme of input assemblies follows a relaxed version of the PanSN prefix naming specification (https://github.com/pangenome/PanSN-spec). The input assembly identifier should consist of three parts: sample tag, delimiter, and haplotype tag. The sample tag and haplotype tag can be any strings consisting of letters and/or numbers, whereas the delimiter should be a user-specified character (option: --sep) that is not used in the assembly/contig names (e.g., # or :). The maximum number of assemblies allowed by VRPG is 65535.

Input pangenome graph specification

In a pangenome graph, a node represents a directed genomic segment, whereas an edge represents the relative order and direction of the two inter-connected nodes. VRPG natively works with pangenome graph files in the rGFA format, which can be straightforwardly constructed using Minigraph (Li et al. 2020). Compared with a pangenome graph encoded in other formats, the rGFA-formatted pangenome graph inherently tracks the origin of each node in terms of its location and direction from the input assembly, which comes in handy when analyzing and interpreting the corresponding graph. In an rGFA-formatted graph, all nodes from the predefined primary linear reference are labeled as rank-0 (defined with the SR tag), whereas the incrementally added nonprimary linear reference nodes derived from other graph-constituent assemblies are labeled with lower ranks (e.g., 1, 2, etc.). VRPG can readily use this information for organizing the displayed graph layouts during graph visualization. Although the rGFA-graph does not store the assembly-to-graph path-traversing information, such information can be straightforwardly recovered by aligning each graph-constituent assembly to the rGFA-graph using Minigraph. In addition to the rGFA-encoded pangenome graphs, VRPG also supports GFAv1-encoded pangenome graphs with a dedicated module (gfa2view) for format transformation.

GFAv1 format transformation for VRPG

GFAv1 is a widely used pangenome graph format. Unlike the rGFA-formatted pangenome graph, there is no predefined primary linear reference for establishing the coordinate system in GFAv1-encoded graphs. Therefore, to accommodate a GFAv1-formated graph with VRPG, a user-defined reference genome is required. When there are multiple copies of highly similar sequence segments (e.g., in the case of segmental or tandem duplication) in the same genome assembly, some pangenome graph builders such as PGGB tend to collapse them into a single node, which is more efficient in compressing genomic information but also breaks the linearity of the reference genome coordinate system. To work around this challenge, the gfa2view module of VRPG will traverse the reference path to find the nodes that are traversed for two or more times (i.e., the collapsed nodes) and insert the corresponding "shadow nodes" and their associated edges into the graph to restore the original reference coordinates for the given node. In this way, the primary linear reference genome rendered by VRPG still keeps linearity. The resulting graph with these added shadow nodes is recorded in gfa2view's output file, which can be used for VRPG as the input file for visualization. Meanwhile, the predefined path (the P-line) and walk (the W-line) information in the GFA file will be extracted by VRPG for tracing the assembly-to-graph path of each graph-constituent assembly. The theoretical maximal node allowance in VRPG is 2,147,483,647. In practice, this upper limit will be lower for PGGB graphs, given that newly introduced "shadow nodes" consume available node indices. There is no upper limit for edge counts in VRPG.

Node and edge rendering

Representative genomic segments are denoted by graph nodes and further illustrated as colored blocks in the view window. For nodes representing the primary linear reference genome (predefined when building the reference pangenome graph), their block sizes are generally proportional to the actual size of the corresponding genomic segments. For nodes representing nonreference assemblies, their corresponding blocks can be optionally scaled. If the length of a node (denoted as "S" herein) is <2 kb, the node will be illustrated as a rectangular block. Otherwise, the node will be represented by a curved block in which the corner count of the curved block is proportional to the multiples of 2 kb determined by S. The connections between different nodes are represented by graph edges and are further illustrated as directed lines. For the cleanness and efficacy of rendering, additional graph simplification algorithms are further applied to trim off those peripheral nodes that are far away from the reference nodes (e.g., when traversing depths are greater than or equal to user-defined search depth) as well as to optionally hide small nodes (e.g., those representing genomic variants that are <50 bp) in the graph. These features are especially helpful when visualizing genome graphs built by Minigraph-Cactus and PGGB in which all base-level variants (e.g., SNVs and indels) are denoted as nodes. In practice, we recommend a maximal query region size of 1 Mb to restrain the total number of nodes and edges for rendering, so that the web browser will not get overwhelmed during graph rendering.

Efficient access to the graph

VRPG implements a block index system to enable quick access to the subgraph corresponding to any given region along the predefined linear reference genome. Briefly, the predefined linear reference genome is first subdivided into blocks, with each block containing 2000 reference nodes. For each block, a breadth-first search algorithm was implemented to find all edges and

nonreference nodes associated with the reference nodes within the corresponding block, and a block index corresponding to these edges and nodes was created. To reduce the search space and accelerate the index building process, by default, given a predefined primary linear reference chromosome, VRPG considers nodes that fall into the following two scenarios: (1) nodes with at least one edge directly connecting to the focal chromosome and (2) nodes with edges that neither are directly connected to the focal chromosome nor are private to any other chromosome. Based on the indexed nodes and edges, an ordered block index array for each assembly path was created. With such a block index system, VRPG can efficiently locate the subgraph associated with the query region (either directly overlapping or connected by edges) for rendering. VRPG uses the "--xDep" parameter (specified via vrpg_preprocess.py or gfa2view) to set the hard maximal search depth allowance for graph traversing during block indexing. When setting this value large enough (default: 100), VRPG should be able to exhaustively retrieve all potentially relevant material for rendering. Note that another "search depth" option (default: 10) is also provided to set the soft maximal search depth allowance for the final graph rendering, which is inherently constrained by the hard limit set by "--xDep". In the case of a common node shared by multiple blocks or even chromosomes within the specified search depth allowance, such association will be recorded for all related blocks and chromosomes so that the node can always be accessed via its associated blocks and chromosomes.

Yeast reference pangenome graph construction

The yeast reference pangenome graph was constructed for this study. This graph was built upon 163 yeast genome assemblies from 142 strains, with some heterozygous/polyploid strains having both phased and collapsed assemblies. Briefly, we took the SGDref (version: R64-1-1) retrieved from the *Saccharomyces* genome database (SGD; https://www.yeastgenome.org/) as well as 162 assemblies from our recently released ScRAP (O'Donnell et al. 2023; Miao et al. 2024) to construct the reference pangenome graph. Minigraph (Li et al. 2020) was used for this graph construction with the command "minigraph -cxggs -l 5000." With the SGDref as the reference genome, we incrementally added those 162 ScRAP assemblies into the graph according to their phylogenetic distances to SGDref. The phylogenetic distances employed here were extracted from the phylogenetic tree of these 163 input genomes built upon their concatenated one-to-one orthologous gene matrix previously described (O'Donnell et al. 2023). Regarding the haplotype tag, for the yeast reference pangenome graph, we used "HP0" to denote haplotypes of haploid or homozygous diploid strains, while using "collapsed," "HP1," and "HP2" to denote collapsed, or the two phased haplotypes of heterozygous diploid strains. The constructed pangenome graph is provided in Supplemental Materials (Supplemental Data File) and has also been deposited at Zenodo (see "Software availability").

Human reference pangenome graph construction

The HPRC constructed three human pangenome graphs using Minigraph, Minigraph-Cactus, and PGGB, respectively, based on 90 genome assemblies from 46 samples (Liao et al. 2023). In addition to the classic (GRCh38) and T2T (CHM13) human reference genome assemblies, the other 44 samples all have two fully phased genome assemblies. For the Minigraph graph, we used the version deposited by Dr. Heng Li with the download link https://zenodo.org/records/6983934/files/GRCh38-90c.r518.gfa.gz. For the Minigraph-Cactus graph, we used the HPRC v1.1 version, accessible via the download link https://s3-us-west-2.amazonaws.com/human-pangenomics/pangenomes/freeze/freeze1/minigraph-cactus/hprc-v1.1-mc-grch38/hprc-v1.1-mc-grch38.gfa.gz. For the PGGB graph, we used the HPRC v1.0 version, accessible via the download link https://s3-us-west-2.amazonaws.com/human-pangenomics/pangenomes/freeze/freeze1/pggb/hprc-v1.0-pggb.gfa.gz. For these human reference pangenome graphs, the haplotype tag "0" was used to denote haplotypes of haploid samples or those for which the haplotype is indeterminate owing to mosaicism or collapsing, whereas "mat" and "pat" were used to denote the maternal and paternal haplotypes of the phased diploid samples. Note that because of the disk space limitation of our web server, we opted for not storing the node sequences for the human pangenome graphs on our demonstration site. Therefore, when clicking on a node in the demonstrated human pangenome graphs, the node sequence will not be reported in the node information panel, which is not the case for the demonstrated yeast pangenome graph.

Annotation track preparation

For the annotation tracks displayed with the yeast pangenome graph, the SGDref's gene, centromere, and telomere annotations were extracted from the NCBI RefSeq GFF3 file (https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/146/045/GCF_000146045.2_R64/GCF_000146045.2_R64_genomic.gff.gz). The subtelomere annotation was extracted from the GFF3 file retrieved from the ScRAPdb database (Miao et al. 2024; https://www.evomicslab.org/db/ScRAPdb/). The GC% was calculated with 250 bp nonoverlap windows using the profile_GC_sliding_window.pl script (options: -w 250 -s 250) from RecombineX (Li et al. 2022). The conserved elements identified based on the whole-genome alignment of seven budding yeast species (phastConsElements7way) were retrieved from the table browser tool provided by the UCSC Genome Browser (Raney et al. 2024).

For the annotation tracks displayed with the human pangenome graph, the GRCh38-based gene annotation was extracted from the NCBI RefSeq GFF3 file (https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/405/GCF_000001405.40_GRCh38.p14/GCF_000001405.40_GRCh38.p14_genomic.gff.gz). The centromeres, Genome Aggregation Database-SVs (gnomAD_SV), and conserved elements identified based on the whole-genome alignment of 100 vertebrate species (phastConsElements100way) were retrieved from UCSC Genome Browser's table browser as described above. For the "high priority clinical genes" track, a previously curated gene set with high-priority clinical significance (merged set: 857 genes) was obtained from the literature (Wagner et al. 2022), and their corresponding genome coordinates were extracted based on the NCBI gene annotation GFF3 file. The GC% was calculated with 2500 bp nonoverlap windows using the aforementioned profile_GC_liding_window.pl script (options: -w 2500 -s 2500).

Software implementation and web deployment

The backend of VRPG is implemented in C++ and Python3 based on the Django framework (https://www.djangoproject.com/), whereas D3.js (https://github.com/d3/d3) was employed at the frontend for data visualization. Bootstrap and jQuery were further used for interactive query and rendering. The demonstration webserver is deployed via an Alibaba Simple Application Server with 2 CPU, 4 GB RAM, and 280 GB ESSD data storage. VRPG (v0.1.5) was used for this web demonstration.

Linear genome comparison and visualization for the demonstrated examples

The genomic regions of the demonstrated loci (the Chr XIV flip/flop inversion and *DOG2* for yeast as well *DSCAM* and *CR1* for

human) were extracted from the corresponding linear genome assemblies and subsequently compared and plotted with EasyFig (GitHub commit: 639daec) (Sullivan et al. 2011).

Pangenome graph visualization benchmarking

We systematically evaluated the graph parsing and rendering performance of VRPG (v0.1.5), Bandage (v0.9.0), GfaViz (v1.0), and PanGraphViewer (v1.0.2; desktop version). For graph parsing, the full-scale yeast and human pangenome graphs as previously described were used. Additionally, the human Chr 22 subgraphs from all three full-scale human pangenome graphs were further extracted by GFAtools (v0.5-r292-dirty; for pangenome graphs built from Minigraph and Minigraph-Cactus) and ODGI (v0.9.0-0-g1895f496; for pangenome graphs built from PGGB). For graph rendering, whereas VRPG and PanGraphViewer can directly access the corresponding subgraph based on query coordinates, Bandage and GfaViz can only extract a subgraph based on queried node IDs. Therefore, to better gauge rendering time and peak memory usage, we prepared the subgraphs for our benchmarked regions with GFAtools and ODGI as described above. The genomic regions for such subgraph extraction include the SGDref-based Chr IV: 1–50,000 and Chr IV: 1–500,000 regions for yeast and the GRCh38-based Chr 22, Chr 22: 20,000,001–20,100,000, and Chr 22: 20,000,001–20,500,000 regions for humans.

All benchmark tests were performed using a desktop computer equipped with Intel Core i5-9400 CPU, 16 GB RAM, and 1 TB hard disk and run with the Ubuntu 24.04.1 LTS operating system. To track the peak memory consumption, the GNU time (v1.9; option: -v; https://www.gnu.org/software/time/) was used for the tests of Bandage, GfaViz, and PanGraphViewer. For VRPG, given its web-based nature, Chrome's (v130.0.6723.91) task manager function was used for assessing the memory consumption. To track the graph parsing and rendering time, the stopwatch function of iPhone's Clock app (v1.1) was used for Bandage. For GfaViz and PanGraphViewer, their natively reported graph parsing and rendering time was used. For VRPG, Chrome's developer tools function was used to calculate the browser's response time.

Software availability

VRPG is free for use under the MIT license, with the source code available in both Supplemental Code File and GitHub (https://github.com/codeatcg/VRPG). The real-case demonstration of VRPG on yeast and human pangenome graphs is hosted at https://www.evomicslab.org/app/vrpg/. The 163-yeast-genome-based pangenome graph constructed for this study is available in both the Supplemental Data as well as at Zenodo (https://zenodo.org/records/13968346/files/ScRAP_v20230121_163asm.minigraph.gfa.gz).

Competing interest statement

The authors declare no competing interests.

Acknowledgments

We thank the valuable comments and suggestions from the three anonymous reviewers, which helped to significantly improve the quality of this manuscript and the associated software. We thank the technical support from Huihui Li and Ludong Yang for software testing and benchmarking. We thank Jing Li for critically reading the manuscript. This work is supported by the National Natural Science Foundation of China (32470663 and 32070592 to J.-X.Y.), Guangdong Pearl River Talents Program (2019QN01Y183 to J.-X.Y.), Fundamental Research Funds for the Central Universities of Sun Yat-sen University (24qnpy293 to J.-X.Y.), and Young Talents Program of Sun Yat-sen University Cancer Center (YTP-SYSUCC-0042 to J.-X.Y.). The funding agencies have not played any role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author contributions: J.-X.Y. conceived the study. Z.M. developed the software. Z.M. and J.-X.Y. analyzed the results. J.-X.Y. and Z.M. wrote the paper. Both authors reviewed and contributed to the final version of the paper.

References

Audano PA, Sulovari A, Graves-Lindsay TA, Cantsilieris S, Sorensen M, Welch AE, Dougherty ML, Nelson BJ, Shah A, Dutcher SK, et al. 2019. Characterizing the major structural variant alleles of the human genome. *Cell* **176:** 663–675.e19. doi:10.1016/j.cell.2018.12.019

Beyer W, Novak AM, Hickey G, Chan J, Tan V, Paten B, Zerbino DR. 2019. Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics* **35:** 5318–5320. doi:10.1093/bioinformatics/btz597

Brouwers N, Van Cauwenberghe C, Engelborghs S, Lambert J-C, Bettens K, Le Bastard N, Pasquier F, Montoya AG, Peeters K, Mattheijssens M, et al. 2012. Alzheimer risk associated with a copy number variation in the complement receptor 1 increasing C3b/C4b binding sites. *Mol Psychiatry* **17:** 223–233. doi:10.1038/mp.2011.24

Chin C-S, Behera S, Khalak A, Sedlazeck FJ, Sudmant PH, Wagner J, Zook JM. 2023. Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. *Nat Methods* **20:** 1213–1221. doi:10.1038/s41592-023-01914-y

De Coster W, Weissensteiner MH, Sedlazeck FJ. 2021. Towards population-scale long-read sequencing. *Nat Rev Genet* **22:** 572–587. doi:10.1038/s41576-021-00367-3

Defenouillère Q, Verraes A, Laussel C, Friedrich A, Schacherer J, Léon S. 2019. The induction of HAD-like phosphatases by multiple signaling pathways confers resistance to the metabolic inhibitor 2-deoxyglucose. *Sci Signal* **12:** eaaw8000. doi:10.1126/scisignal.aaw8000

Eizenga JM, Novak AM, Sibbesen JA, Heumos S, Ghaffaari A, Hickey G, Chang X, Seaman JD, Rounthwaite R, Ebler J, et al. 2020. Pangenome graphs. *Annu Rev Genomics Hum Genet* **21:** 139–162. doi:10.1146/annurev-genom-120219-080406

Garrison E, Guarracino A. 2023. Unbiased pangenome graphs. *Bioinformatics* **39:** btac743. doi:10.1093/bioinformatics/btac743

Garrison E, Guarracino A, Heumos S, Villani F, Bao Z, Tattini L, Hagmann J, Vorbrugg S, Marco-Sola S, Kubica C, et al. 2024. Building pangenome graphs. *Nat Methods* **21:** 2008–2012. doi:10.1038/s41592-024-02430-3

Gonnella G, Niehus N, Kurtz S. 2019. *GfaViz*: flexible and interactive visualization of GFA sequence graphs. *Bioinformatics* **35:** 2853–2855. doi:10.1093/bioinformatics/bty1046

Grossman TR, Gamliel A, Wessells RJ, Taghli-Lamallem O, Jepsen K, Ocorr K, Korenberg JR, Peterson KL, Rosenfeld MG, Bodmer R, et al. 2011. Over-expression of DSCAM and COL6A2 cooperatively generates congenital heart defects. *PLoS Genet* **7:** e1002344. doi:10.1371/journal.pgen.1002344

Guarracino A, Heumos S, Nahnsen S, Prins P, Garrison E. 2022. ODGI: understanding pangenome graphs. *Bioinformatics* **38:** 3319–3326. doi:10.1093/bioinformatics/btac308

Gustafson JA, Gibson SB, Damaraju N, Zalusky MPG, Hoekzema K, Twesigomwe D, Yang L, Snead AA, Richmond PA, Coster WD, et al. 2024. High-coverage nanopore sequencing of samples from the 1000 Genomes Project to build a comprehensive catalog of human genetic variation. *Genome Res* **34:** 2061–2073. doi:10.1101/gr.279273.124

Hickey G, Monlong J, Ebler J, Novak AM, Eizenga JM, Gao Y, Human Pangenome Reference Consortium; Marschall T, Li H, Paten B. 2024. Pangenome graph construction from genome alignments with Minigraph-Cactus. *Nat Biotechnol* **42:** 663–673. doi:10.1038/s41587-023-01793-w

Jiao W-B, Schneeberger K. 2020. Chromosome-level assemblies of multiple *Arabidopsis* genomes reveal hotspots of rearrangements with altered evolutionary dynamics. *Nat Commun* **11:** 989. doi:10.1038/s41467-020-14779-y

Kucukkilic E, Brookes K, Barber I, Guetta-Baranes T, Morgan K, Hollox EJ, ARUK Consortium. 2018. Complement receptor 1 gene (*CR1*) intragenic duplication and risk of Alzheimer's disease. *Hum Genet* **137:** 305–314. doi:10.1007/s00439-018-1883-2

Lambert J-C, Heath S, Even G, Campion D, Sleegers K, Hiltunen M, Combarros O, Zelenika D, Bullido MJ, Tavernier B, et al. 2009.

Genome-wide association study identifies variants at CLU and CR1 associated with Alzheimer's disease. *Nat Genet* **41:** 1094–1099. doi:10.1038/ng.439

Li H, Feng X, Chu C. 2020. The design and construction of reference pangenome graphs with minigraph. *Genome Biol* **21:** 265. doi:10.1186/s13059-020-02168-z

Li J, Llorente B, Liti G, Yue J-X. 2022. Recombinex: a generalized computational framework for automatic high-throughput gamete genotyping and tetrad-based recombination analysis. *PLoS Genet* **18:** e1010047. doi:10.1371/journal.pgen.1010047

Lian Q, Huettel B, Walkemeier B, Mayjonade B, Lopez-Roques C, Gil L, Roux F, Schneeberger K, Mercier R. 2024. A pan-genome of 69 *Arabidopsis thaliana* accessions reveals a conserved genome structure throughout the global species range. *Nat Genet* **56:** 982–991. doi:10.1038/s41588-024-01715-9

Liao W-W, Asri M, Ebler J, Doerr D, Haukness M, Hickey G, Lu S, Lucas JK, Monlong J, Abel HJ, et al. 2023. A draft human pangenome reference. *Nature* **617:** 312–324. doi:10.1038/s41586-023-05896-x

Liu H, Caballero-Florán RN, Hergenreder T, Yang T, Hull JM, Pan G, Li R, Veling MW, Isom LL, Kwan KY, et al. 2023. DSCAM gene triplication causes excessive GABAergic synapses in the neocortex in down syndrome mouse models. *PLoS Biol* **21:** e3002078. doi:10.1371/journal.pbio.3002078

Miao Z, Ren Y, Tarabini A, Yang L, Li H, Ye C, Liti G, Fischer G, Li J, Yue J-X. 2024. ScRAPdb: an integrated pan-omics database for the *Saccharomyces cerevisiae* reference assembly panel. *Nucleic Acids Res* **53:** D852–D863. doi:10.1093/nar/gkae955

Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze AV, Mikheenko A, Vollger MR, Altemose N, Uralsky L, Gershman A, et al. 2022. The complete sequence of a human genome. *Science* **376:** 44–53. doi:10.1126/science.abj6987

O'Donnell S, Yue J-X, Saada OA, Agier N, Caradec C, Cokelaer T, De Chiara M, Delmas S, Dutreux F, Fournier T, et al. 2023. Telomere-to-telomere assemblies of 142 strains characterize the genome structural landscape in *Saccharomyces cerevisiae*. *Nat Genet* **55:** 1390–1399. doi:10.1038/s41588-023-01459-y

Paten B, Novak AM, Eizenga JM, Garrison E. 2017. Genome graphs and the evolution of genome inference. *Genome Res* **27:** 665–676. doi:10.1101/gr.214155.116

Raney BJ, Barber GP, Benet-Pagès A, Casper J, Clawson H, Cline MS, Diekhans M, Fischer C, Navarro Gonzalez J, Hickey G, et al. 2024. The UCSC Genome Browser database: 2024 update. *Nucleic Acids Res* **52:** D1082–D1088. doi:10.1093/nar/gkad987

Rech GE, Radío S, Guirao-Rico S, Aguilera L, Horvath V, Green L, Lindstadt H, Jamilloux V, Quesneville H, González J. 2022. Population-scale long-read sequencing uncovers transposable elements associated with gene expression variation and adaptive signatures in *Drosophila*. *Nat Commun* **13:** 1948. doi:10.1038/s41467-022-29518-8

Salzberg LI, Martos AAR, Lombardi L, Jermiin LS, Blanco A, Byrne KP, Wolfe KH. 2022. A widespread inversion polymorphism conserved among Saccharomyces species is caused by recurrent homogenization of a sporulation gene family. *PLoS Genet* **18:** e1010525. doi:10.1371/journal.pgen.1010525

Sullivan MJ, Petty NK, Beatson SA. 2011. Easyfig: a genome comparison visualizer. *Bioinformatics* **27:** 1009–1010. doi:10.1093/bioinformatics/btr039

Wagner J, Olson ND, Harris L, McDaniel J, Cheng H, Fungtammasan A, Hwang Y-C, Gupta R, Wenger AM, Rowell WJ, et al. 2022. Curated variation benchmarks for challenging medically relevant autosomal genes. *Nat Biotechnol* **40:** 672–680. doi:10.1038/s41587-021-01158-1

Weller CA, Andreev I, Chambers MJ, Park M, Program NCS, Bloom JS, Sadhu MJ, Barnabas BB, Black S, Bouffard GG, et al. 2023. Highly complete long-read genomes reveal pangenomic variation underlying yeast phenotypic diversity. *Genome Res* **33:** 729–740. doi:10.1101/gr.277515.122

Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* **31:** 3350–3352. doi:10.1093/bioinformatics/btv383

Yang X, Wang X, Zou Y, Zhang S, Xia M, Fu L, Vollger MR, Chen N-C, Taylor DJ, Harvey WT, et al. 2023. Characterization of large-scale genomic differences in the first complete human genome. *Genome Biol* **24:** 157. doi:10.1186/s13059-023-02995-w

Yokoyama TT, Sakamoto Y, Seki M, Suzuki Y, Kasahara M. 2019. MoMI-G: modular multi-scale integrated genome graph browser. *BMC Bioinformatics* **20:** 548. doi:10.1186/s12859-019-3145-2

Yuan Y, Ma RK-K, Chan T-F. 2023. PanGraphViewer: a versatile tool to visualize pangenome graphs. bioRxiv doi:10.1101/2023.03.30.534931

Yue J-X, Li J, Aigrain L, Hallin J, Persson K, Oliver K, Bergström A, Coupland P, Warringer J, Lagomarsino MC, et al. 2017. Contrasting evolutionary genome dynamics between domesticated and wild yeasts. *Nat Genet* **49:** 913–924. doi:10.1038/ng.3847

Received April 13, 2024; accepted in revised form January 8, 2025.