

Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes

Received: 10 August 2022

Accepted: 17 May 2023

Published online: 26 June 2023

 Check for updates

Chen-Shan Chin ^{1,2}✉, Sairam Behera ³, Asif Khalak ², Fritz J. Sedlazeck ^{3,4}, Peter H. Sudmant ⁵, Justin Wagner⁶ & Justin M. Zook ⁶

Advancements in sequencing technologies and assembly methods enable the regular production of high-quality genome assemblies characterizing complex regions. However, challenges remain in efficiently interpreting variation at various scales, from smaller tandem repeats to megabase rearrangements, across many human genomes. We present a PanGenome Research Tool Kit (PGR-TK) enabling analyses of complex pangenome structural and haplotype variation at multiple scales. We apply the graph decomposition methods in PGR-TK to the class II major histocompatibility complex demonstrating the importance of the human pangenome for analyzing complicated regions. Moreover, we investigate the Y-chromosome genes, *DAZ1/DAZ2/DAZ3/DAZ4*, of which structural variants have been linked to male infertility, and X-chromosome genes *OPN1LW* and *OPN1MW* linked to eye disorders. We further showcase PGR-TK across 395 complex repetitive medically important genes. This highlights the power of PGR-TK to resolve complex variation in regions of the genome that were previously too complex to analyze.

Studying genomes, the fundamental information contained in all living beings, is the foundation for understanding the biology and evolution of all organisms, as well as the genetic diseases of humans. Despite the millions of human genomes that have been sequenced since the onset of the Human Genome Project^{1,2} and the dramatic reduction in the cost of short-read (roughly 150 bp) DNA sequencing, there is still fundamental information yet to be revealed in genomics³. While it is important to recognize successes so far, including small variant surveys, genome-wide association studies^{4–7} and the development of routine laboratory tests for genetic-based precision medicine^{8–11}, there remain fundamental biological questions that involve structures at greater length scales

that can only be captured using long-range information accessible by long-read technologies and diploid phased assemblies^{12–14}.

With the possibility of resolving variants at multiple scales, small and large, researchers now can fully characterize previously inaccessible regions by focusing on single-nucleotide polymorphisms and small indels alone^{15,16}. Examples of such previously inaccessible regions include centromere, telomeres and complex repeat regions. Recent results with pangenome-scale de novo human assemblies and the CHM13 telomere to telomere assembly have already shown the potential for revealing biological insights^{3,17–20}, which are the foundation for understanding complex genetic diseases.

¹GeneDX, Stamford, CT, USA. ²Foundation of Biological Data Science, Belmont, CA, USA. ³Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX, USA. ⁴Department of Computer Science, Rice University, Houston, TX, USA. ⁵Department of Integrative Biology, University of California Berkeley, Berkeley, CA, USA. ⁶Material Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA.

✉e-mail: jchin@genedx.com

A concept that becomes powerful in such analyses is that of the pangenome: that is, a characterization of both the genetic structure and the genetic variation across diverse individuals of a species. However, such complexity and diversity generate interpretive challenges that require more advanced tools. A graph representing many genome assemblies at once provides a way to visualize and analyze complicated structural variations among different haplotypes^{21–28}. Previously, distinct approaches to generate graphs representing pangenome structures have been proposed for various applications. For example, variant graph^{27,29} and PanGenie²⁶ focus on improving variant calling and genotyping with pangenome references. Cactus graphs²⁵, Progressive Cactus graphs³⁰, PGGB¹⁹ and cactus-mingraph^{19,24} build pangenome graphs aiming for large-scale structural rearrangement comparisons. Stringomics graph with ‘stringlet’²¹, Seqwish²² and de Bruijn graph based approaches^{26,31} provide algorithms and data structures for improving storage and query efficiency and reduce bias caused by the alignment processes. These tools provide more accessible pictures for researchers to understand repeats and rearrangements than using computational intensive and visually complicated multiple sequence alignments (MSA)^{30,32,33}. The traditional MSA view is typically represented as a big matrix where each row represents a different genome, and the columns represent the bases. With MSA, the relationships between sequences are not obvious when there are complicated repeats or structural variations. Instead of per base alignment, a pangenome graph effectively condenses the homologous regions and can express the relationship between those different regions through graph edge connections that are easier to trace. Meanwhile, although a graph is an elegant data structure for gathering information from pangenomic assemblies, there remains a gap in projecting the underlying linear sequences onto a graph at various scales to reveal and compare features of many different haplotypes²³.

To address this gap, we present a generalized graph framework as a software package, PanGenome Research Tool Kit (PGR-TK, <https://github.com/GeneDx/pgr-tk>), that is scalable to rapidly represent multiple samples at varying resolution levels by adopting different parameters to facilitate exploratory analysis. PGR-TK is able to resolve and visualize the most complex regions of the human genome that often affect multiple medical important phenotypes (for example, *LPA*, *HLA* and so on).

We demonstrate the ability of PGR-TK to visualize and enable deeper insights into complex variants in repetitive genes, including a gene within nested palindromic and tandem repeats (*AMY1A*), the major histocompatibility complex (MHC) region including the complex human leukocyte antigen (HLA) class II locus³⁴, the Genome in a Bottle (GIAB) challenging medically relevant gene list³⁵ and chrX and chrY ampliconic genes³⁶. Many genome-wide studies including GIAB have excluded many of these genes from their analyses because they are difficult to represent in VCF and it is challenging to compare differing representations³⁷. To understand how PGR-TK can help with the challenge of variant calling, variant representation and comparison across these genes and genomic loci we use the Human Pangenome Reference Consortium (HPRC)^{19,38} year one 47 human genome assemblies (94 diverse haplotypes). With the ability to survey a large set of genes swiftly with PGR-TK, we hope to understand how to better provide a broader benchmark set for challenging genes using HPRC assemblies in the future. We examine *OPN1LW* and *OPN1MW* on chromosome X and *DAZI2/3/4* on chromosome Y in detail to understand how the limit due to complicated large-scale genome rearrangement affects the current methodology of generating variant call benchmarks. Our initial analysis of the GIAB clinical and medically important genes (CMRG) with a pangenome graph approach will help the research community to adapt the pangenome resource for clinical and medical genetic applications. Tools for visualizing and analyzing complicated rearrangement loci such as PGR-TK will be essential for better variant

calling and understanding how structural variations with repeats affect the results for the community.

Results

PGR-TK

The PGR-TK has several different components to facilitate rapid pangenome analysis. The general scope and design of the PGR-TK is illustrated in Fig. 1a. PGR-TK applies the computation techniques and data structures initially developed for fast genome assemblers^{39–41} to pangenome analysis tasks. Instead of building a whole genome graph at once, which can be computationally expensive. PGR-TK provides tools for building an indexed sequence database, fetching and querying sequences of interest (for example, genes or regions with large-scale structural variations) from the database to create pangenomics graphs accordingly. It uses minimizer anchors to generate pangenome graphs at different scales without more computationally intensive sequence-to-sequence alignment or explicitly calling variants with respect to a reference. The generation step of the pangenome graph considers all input sequences equivalently without a preferential reference. Note that the sequence fetching step using a query sequence may introduce bias due to missing or incorrect alignments. We also developed an algorithm to decompose tangled pangenome graphs to more manageable units (principal bundles). With such decomposition, we can easily project the linear genomics sequence onto the principal bundles. It can provide more straightforward visualization to generate insight by revealing the contrast of the repeat and rearrangement variations among the haplotypes. Such pangenome-level graph decomposition provides utilities similar to the A-de Bruijn graph approach for identifying repeats and conserved segmental duplications^{42–45}, but for the whole human pangenome collection at once.

PGR-TK uses the Assembly Genome Compressor⁴⁶ for storing pangenome assembly contigs and includes binary for creating the sparse hierarchical minimizer (SHIMMER) index. For the HPRC year one data release (94 fully assembled haplotypes from 47 samples), it takes 18 minutes to create the index file on an Amazon Web Services c5.12xlarge instance, with the default parameters. This is substantially faster than building an alignment index for query with tools such as mimimap2 (Supplementary Table 1). Although PGR-TK was designed to retrieve homologous sequences from the database, rather than finding the best alignments, our evaluations indicate that the query results are generally consistent with other alignment tools (Supplementary Tables 2 and 3).

Once the index is built, it can be loaded into memory within minutes. As shown in Fig. 1a, there are three main functional modules using the index: (1) fetching homologous regions and sequences of the pangenome database given a query sequence, (2) creating a minimizer anchored pangenome-graph (MAP-graph) and (3) command line tools and a software library for interactive analysis and visualization on the generated graph and the underlying sequences. One of the main applications of PGR-TK is for deconvolving large regions of the human genome to reveal complex variations. It offers a set of efficient command line tools for various tasks, but also allows for more interactive and in-depth analysis through its integration with the Jupyter Laboratory⁴⁷ and other data science tools. This makes it a valuable resource for researchers seeking to uncover insights from their genomic data.

The source code and library can be downloaded from <https://github.com/GeneDX/pgr-tk>. The documentation of the Python APIs is at <https://genedx.github.io/pgr-tk/>.

SHIMMER index

SHIMMER is a data structure extending the minimizer for more efficient indexing over larger regions. Additional minimizer reduction steps to generate sparse minimizers are applied to the minimizer sequences instead of the original base-pair sequences in a hierarchical way³⁹. Such sparse minimizers can serve as natural ‘anchors’ or ‘markers’

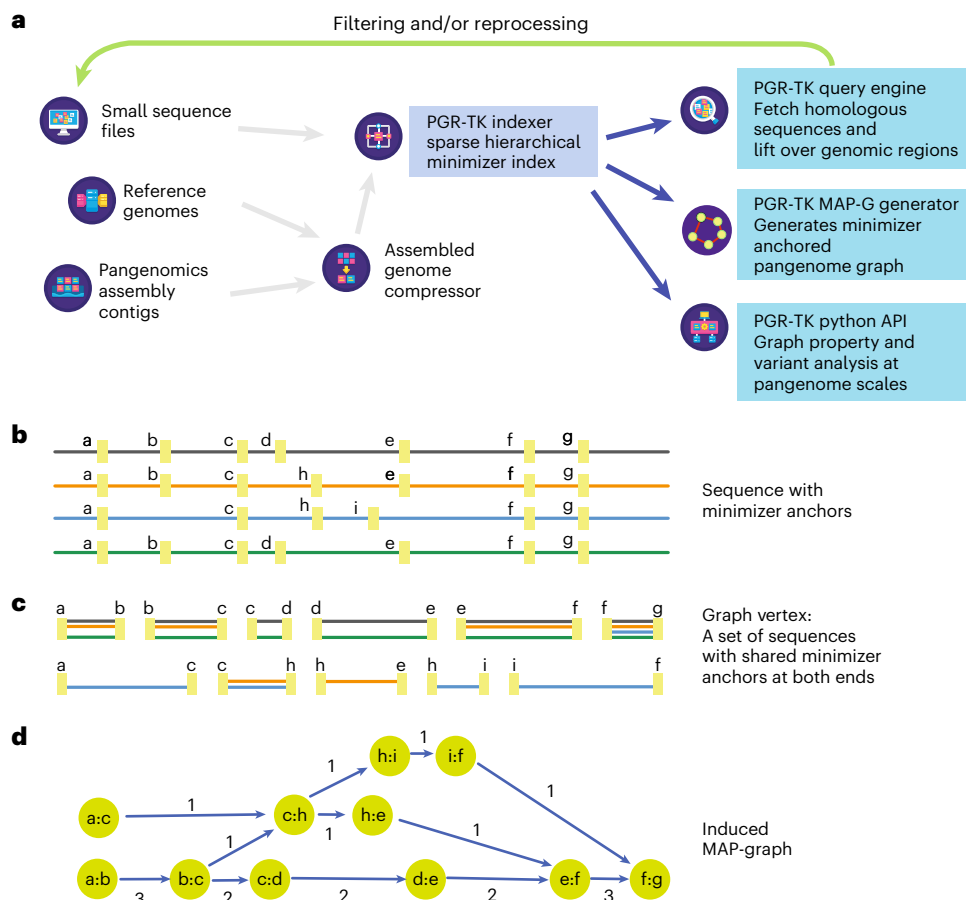


Fig. 1 | The architecture of PGR-TK and minimizer anchored graph construction. **a**, Overall architecture and design scope of the PGR-TK library. **b**, Each sequence in the database is scanned, and the location of the minimizers are recorded to construct the SHIMMER database and MAP-graph. **c**, Each vertex

in the MAP-graph represents a collection of sequence fragments sharing the two ending SHIMMERS in the database. **d**, The MAP-graph is constructed by merging all paths from all sequences into a graph.

on genomics sequences without an explicit reference coordinate system. We use the SHIMMERS for quick sequence queries as initially proposed by Roberts et al.⁴⁸. PGR-TK identifies all neighboring pairs of SHIMMERS and indexes all the sequence segments between the pairs. Figure 1b shows a cartoon of the SHIMMERS identified on each sequence. Then, the pairs of the neighboring SHIMMERS are used for indexing the corresponding sequence segments within the paired SHIMMERS. After that, we build a look-up table of all pairs of SHIMMERS to all segments with the same pair at both ends (Fig. 1c). For the query, we compute the neighboring SHIMMER pairs from a query sequence and search the database for all segments indexed by the same pairs. Finally, we can fetch all target segments stored in the database to get all related sequence information for further analysis. PGR-TK provides functions to refine the raw query results and filter out spurious alignments likely caused by repeats outside the region of interest. With the set of sequences homologous to the query sequence, we can quickly perform downstream analysis work, for example, variant discovery by aligning the sequences to each other. Furthermore, we can generate a local pangenomics (MAP-graph) for comparing the sequences in the pangenome dataset at various scales by adjusting parameters to fit different analysis tasks.

MAP-graph

PGR-TK provides tools to generate the MAP-graph from a set of homologous sequences. The vertices in a MAP-graph are labeled with the neighboring SHIMMER pairs representing a set of sequence segments

in the database (Fig. 1c,d and Methods). The edges in the MAP-graph are induced when at least one sequence connects the two fragments. Thus, each sequence naturally corresponds to a path in the graph, and the vertices in the path also contain the segments of other sequences in the database that share the same SHIMMER pair label (see the Methods for a precise mathematical definition of a MAP-graph). The successful deployment of minimizer- or minhash-based approaches in sequence comparison^{39,40,49} indicates that sequence segments with the same minimizer labels are also likely to be highly homologous. The homology between sequences can be further confirmed by explicit sequence alignment of the segment inside a MAP-graph vertex. However, the computation intensive base-to-base alignment is not required for building the MAP-graph.

The MAP-graph construction in PGR-TK is highly efficient, as it does not rely on traditional sequence-to-sequence alignment. This is demonstrated by the fast graph construction for a set of 147 MHC class II region sequences from the pangenome reference, which was completed in under 5 seconds of wall-clock time using PGR-TK, compared to the 3.5 minutes using seqwish²² and 13 minutes using minigraph²⁴ (Supplementary Table 4).

The size of vertices in the MAP-graph, which represents the lengths of the sequence segments in the pangenome, can be adjusted by changing the parameters that determine the distance between minimizers. This allows us to study genomic features at different length scales and generate pangenome graphs with varying levels of detail. This is particularly useful when analyzing features that vary in size, such as

tandem repeats in the human genome, which can range from a few hundred base pairs to 1–2 kilobases (Supplementary Table 5). By generating pangenome graphs at different levels of detail, we can gain a more comprehensive understanding of complex variation patterns within populations and focus on specific features of interest.

The analysis of pangenome structure can be adjusted by controlling the parameters of minimizer window size (w), minimizer size (k) and hierarchical reduction factor (r), along with an auxiliary parameter min_span , which sets the minimum distance between minimizers in the construction of the SHIMMER index and MAP-graph (Methods). The length of each vertex in the MAP-graph, representing a sequence segment, can be modified by adjusting these parameters. This allows us to study genomics features at different length scales.

Supplementary Fig. 1 illustrates the vertex length distribution for different parameter sets using chromosome 1 of CHM13 assembly. An increase in either w or r results in longer sequences being represented by each vertex, enabling a sparser sampling of the pangenome. The choice of parameters depends on the length of the region of interest and the size of relevant biological features, such as repeat sizes and distances. For example, when studying large-scale differences, bigger w and r values are preferable to generate a sparse index that can efficiently capture large-scale differences. Conversely, to compare small-scale differences, smaller w or r values should be used. Determining the optimal parameters for the pangenome graph generation step can be challenging if the underlying interesting features are less understood. Considering this, we have found that the best initial parameter choice is determined by the length of the sequences of interest to ensure comprehensible results. Based on our observation, we provide a simple formula for selecting the parameters (Supplementary Table 6, Fig. 2 and Supplementary Fig. 2 for related examples).

Principal bundle decomposition

A pangenome graph can serve as a cornerstone for analyzing repeat structure variation in population^{17,19}. It is usually hard to compare multiple sequences with complicated repeat structure by examining pairwise sequence alignments directly. The traditional visualization technique of the dot plot⁵⁰ allows us to perceive the complexity of the repeats but it does not provide insights into the repeat structures as linear representations across each individual sequence directly. Furthermore, only two sequences can be compared with a dot plot.

As an example, we use PGR-TK to investigate the repeat structure of the *AMY1A* gene (Alpha-amylase 1, an enzyme for the first step of catalyzing starch and glycogen in saliva) locus. We pick *AMY1A* as it has various numbers of copies caused by larger-scale structure variation related to the repeat surrounding the gene. The dot plots from randomly picking 36 sequences of a 400 kb region around *AMY1A* in the first year HPRC assemblies to the GRCh38 *AMY1A* reference sequence are shown in Supplementary Fig. 3a. Visual inspections of the dot plots show there are various numbers of copies of forward and inverted repeats forming palindromic sequences at the scales of 100 kb, and from zero up to five palindrome units. Still, only pairwise comparisons are enabled with dot plots and, thus, we lack a comprehensive assessment.

For comparison, we generate the *AMY1A* MAP-graphs at two different scales (Fig. 2) from the HPRC year one assemblies (47 samples). These can be generated with PRG-TK in less than 3 minutes from indexed sequence data. In addition to the MAP-graph, we provide tools analyzing a MAP-graph to ‘relinearize’ the graph into a set of ‘principal bundles’. We design the algorithm to generate the principal bundles representing those consensus paths that are most likely corresponding to repeat units in the pangenomes. The algorithm searches the paths that most of the pangenome sequences go through without branching as the principal bundles. This is analogous to identifying the contigs^{31,32} in genome assembly algorithms.

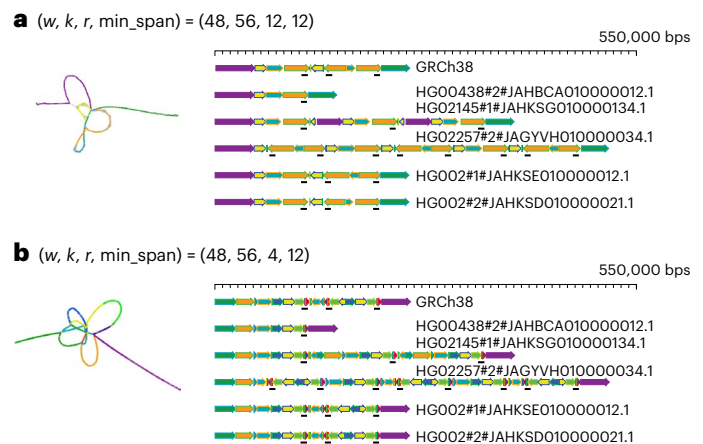


Fig. 2 | *AMY1A* MAP-graph in two different scales. **a**, The left panel shows a sparse MAP-graph representation of the *AMY* region with $(w, k, r, \text{min_span}) = (48, 56, 12, 12)$. 503 vertices and 699 edges represent the 200–550 kb *AMY* region. The graph vertices are colored by the principal bundles that correspond to the principal bundle decomposition of selected genomes on the right panel (gray vertices are those that are not in the principal bundles). **b**, The left panel shows a denser MAP-graph with $r = 4$. The graph has 3,471 vertices and 2,684 edges, which is about 5 times as much as the MAP-graph in **a**. The principal bundle decomposition reveals a more detailed repeat structure than in **a**.

Figure 2 shows the principal bundle decomposition of the *AMY1A* region MAP-graph in two different scales for comparison. The smaller choice of ‘ r ’ generates a MAP-graph with more vertices in the graph and each vertex only represents a smaller portion of the pangenome. This leads to a finer scale of principal bundle decompositions.

The linear representation derived from the MAP-graph allows for efficient identification and classification of repeat structures, which are otherwise challenging to characterize. Seven genomes were selected for analysis in Fig. 2, each demonstrating distinct repeat structures. For example, HG00438 no. 2 mostly lacks repetitive sequences. The GRCh38 one has one relatively simple invert repeat (forming a palindrome region). HG02145 no. 1 has three copies of non-inverted repeats (labeled as repeat 1, 2 and 3). HG02257 no. 2 has three palindromic repeats (labeled as P1, P2 and P3). The two haplotypes from HG002 have similar structures to the GRCh38, except there is an inversion in the middle of the palindrome repeats in one of the two haplotypes. A full plot with all 96 repeat structures, including a hierarchical clustering tree identifying the similarities, is shown in the Supplementary Fig. 3b. This decomposition approach can be used by researchers to effectively classify the repeat structures of regions of interest.

Pangenome analysis of the MHC class II locus

The MHC region in human genomes is highly polymorphic. The genomic sequences of the MHC are fundamental for understanding a human’s adaptive immune system and autoimmune diseases^{53,54}. Owing to its complexity and polymorphic nature, it has been challenging to get a complete picture of the MHC genomics in the human population and to benchmark variant calling in the most variable regions^{34,55}. The HPRC assemblies provide a new opportunity to analyze the MHC sequences with nearly fully assembled sequences of the region.

To showcase the effectiveness of PGR-TK in analyzing complicated human haplotype structures and sequences, we applied it to the HLA class II locus (GRCh38, chr6:32,313,513–32,992,088). We fetched the HPRC HLA class II haplotype sequences by anchoring them with more conserved flanking regions. Our dataset consists of a total of 105 full-length sequences, ranging from 650 to 800 kbp. Figure 3a shows the principal bundle decomposition of the collection of the sequence and Fig. 3b illustrates the MAP-graph of the 105 sequences. The tangled region in the MAP-graph represents highly polymorphic

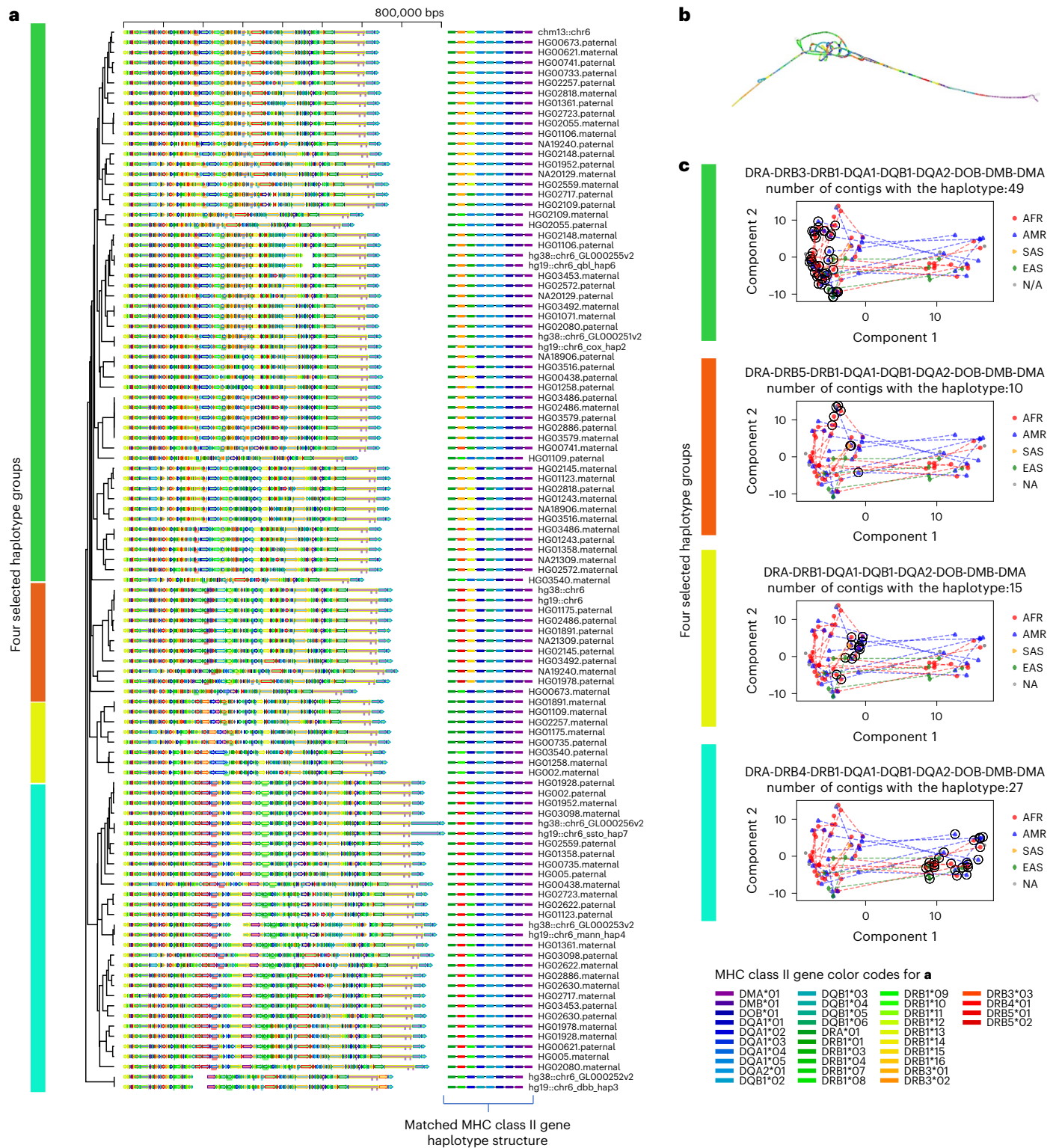


Fig. 3 | Principal bundle decomposition reveals distinct haplotype groups. **a**, The principal bundle decomposition and annotated HLA class II genes in each of the haplotype sequences. The auxiliary tracks below each sequence on the left panel show the locations of the genes. The colors of the auxiliary tracks match the gene list of genes identified for each haplotype on the right. **b**, The MAP-graph generated by PGR-TK. **c**, PCA plots of the MHC class II sequences. Each panel highlights the different gene haplotype combinations. The vertical

color bars indicate the matched haplotype groups in **b** and **c**. The circled symbols indicate the haplotypes belong to the corresponding group. The dotted lines represent the connection between the two haplotypes of individuals included in the analysis set who possess both haplotypes. The population groups, African Ancestry (AFR), American Ancestry (AMR), South Asia Ancestry (SAS), East Asia Ancestry (EAS) and not applicable (NA), are indicated with different markers of different colors.

haplotypes in the human population. We can uncover the combinatorial nature of haplotype variation, such as the appearance of different combinations of HLA class II gene types in different haplotypes, in

relation to the newly released human pangenome references (Fig. 3a) by generating the MAP-graph and the principal bundle decomposition of the MHC class II.

We constructed a hierarchical dendrogram on top of the principal bundle decomposition to study the relationships of highly polymorphic haplotypes in the MHC region of the human population. The PGR-TK provides a command line tool to compute a distance metric derived from pairwise sparse alignment of the bundles between two sequences and generate a dendrogram from all pairwise distances. Known HLA class II gene sequences were also mapped to 105 pangenome sequences. The full annotated principal bundle decomposition, which was annotated with HLA class II genes and the hierarchical clustering dendrogram, is shown in Fig. 3a. The highly polymorphic region in the MAP-graph was found to correspond to the *DRB-1/3/4/5*, *DOB* and *DOA1/2* region. By clustering the full set of haplotype sequences, we can identify the combination of bundles in the entire region that correspond to the gene combinations of each cluster. Our approach uses the principal bundles to classify the complete sequence, rather than relying solely on gene fragments. This classification has the potential to facilitate new applications for improved genotyping or haplotyping of larger population data in this complex region. The results could offer valuable insights into the relationship between haplotype sequences and gene combinations in the context of genetic variation and disease susceptibility.

We can also use the vertices in the MAP-graph to conduct a principal component analysis (PCA) of the MHC class II regions. We collected all vertices in the MAP-graph to form the basis of vectors. Then, we constructed a binary vector for each haplotype path by indicating whether the path of the haplotype passes through the vertex. Figure 3c displays the principal component plot of the haplotype path vectors, along with the ethnic groups. The dotted line connects the two haplotypes of an individual in the sample. We have highlighted four different groups based on the HLA class II gene combinations (excluding the subtypes). Each group contains 10–49 haplotypes. With the current dataset of a limited number of haplotypes, we have not found any statistically significant patterns yet. However, as additional data will be released from HPRC in the coming years, we anticipate that the MAP-graph can be used to systematically analyze this region and better understand its impact on human disease within ethnic group structure.

Analyzing medically relevant amplicon genes

GIAB is using the fully assembled HG002 chrX and chrY from T2T to form new small variant and structural variant benchmarks. The assembly fully resolves the medically relevant ampliconic genes^{36,56,57} *OPN1LW/OPN1MW/OPN1MW2/OPN1MW3* and *DAZ1/DAZ2/DAZ3/DAZ4*, but the variation in these genes is too complex for current approaches to make reliable variant calls compatible with current benchmarking tools.

For example, the genes *OPN1MW* and *OPN1MW2* are inside a 74 kb deletion in HG002 relative to GRCh38, so HG002 contains only two of the four copies of the array in GRCh38: *OPN1LW* and one copy of *OPN1MW/OPN1MW2/OPN1MW3*. Dipcall⁵⁸ can call the 74 kb deletion and variants in the other gene copies, but it may be possible to align the assembly to GRCh38 in alternative ways. The visualization from PGR-TK makes clear the varying number of genes in this array in each haplotype in Fig. 4a, which is important for some phenotypes such as color blindness, since seeing full color requires *OPN1LW* and at least one copy of *OPN1MW/OPN1MW2/OPN1MW3* (refs. 56,59).

Another important gene family *DAZ1/DAZ2/DAZ3/DAZ4* are in a set of nested palindromic repeats. It has been reported that partial deletions in this region may cause male infertility⁵⁷. It would be useful to understand the natural distribution of non-pathogenic structural variants across this ampliconic gene cluster. *DAZ1* and *DAZ2* are roughly 1.5 Mbp (megabasepairs) from *DAZ3* and *DAZ4*, and HG002 has a 1 to 2 Mbp inversion relative to GRCh38 with breakpoints in the segmental duplications that contain the *DAZ* genes (Fig. 4b). In addition to the large inversion, the *DAZ* genes contain structural variants, including a roughly 10 kb deletion in *DAZ2*, two deletions in *DAZ4*

and two insertions in *DAZ3* of sequences that are only in *DAZ1* and *DAZ4* in GRCh38. PGR-TK's ability to color and visualize variation with the principal bundle decomposition algorithm at multiple scales enables intuitive understanding of this type of very complex variation, which would be very difficult to represent and understand as simple structural variant calls in VCF format.

The efficiency of PGR-TK makes it suitable for analyzing complex variants at isolated loci, as well as a set of region of interest. For example, GIAB has identified a set of 395 challenging but medically relevant genes. Analyzing this set in the pangenome references will provide some insight into the related complicated variation at the population level.

Using PGR-TK, we extracted all sequences of the 395 genes from the HPRC year one release (94 haplotype assemblies), CHM13 v.1.1, GRCh38 and hg19 of all 385 CMRG. We generated a MAP-graph for each gene and output it in GFA format (as seen in the Supplementary Fig. 4). For each graph, we derived two metrics to estimate (1) the degree of polymorphism among the pangenomes, and (2) the repeat content considering the variations of the pangenomes (as shown in Fig. 4c). These two measurements provide independent assessments of the MAP-graph structures of these genes. We found, as expected, that highly repetitive genes (such as *LPA* and *KATNAL2*) are more difficult to create a reliable variant benchmark call set. Many highly repetitive genes are excluded from the current CMRG benchmark set. We did not observe a correlation between higher entropy and the reduction of the gene in the benchmark set. We found that the high entropy genes also span larger regions in the genome. While entropy can indicate the complexity of variations in the population, we observed different clustering structures of the top entropy genes (see the comparison of the MAP-graph PCA plots of *SNTG2* and *KMT2C* in Supplementary Fig. 5).

Discussion

With the advance in DNA sequencing technologies, more comprehensive human genomes at, or close to, telomere to telomere will be collected and made available in the coming years. It will enable researchers to study and characterize those previously inaccessible complex, but likely relevant, regions. The current HPRC assembly release has significantly affected our understanding of the human genome architecture. It will also be essential for building applications for clinical and medical tests and diagnostics soon. Flexible and scalable computational tools for analyzing pangenome-level genome assemblies will be part of the vital task of improving the practice of precision medicine with rich genomic data such as those from HPRC.

Many of the recently developed pangenome analysis tools allow graph analysis at the whole-genome level^{22–24}. Meanwhile, the richness of diversity of human genomes over the repetitive regions poses unique challenges for analysis. In our work developing PGR-TK, we focus on providing a flexible library of useful algorithms. Furthermore, it enables analyzing the genome assemblies such that a developer or a researcher can rapidly access certain complex regions by adjusting parameters for visualization and integrating with subsequent analysis. PGR-TK may not be suitable for analysis at the whole-genome scale yet, and obtaining optimized results with the tool might not be straightforward. The challenges associated with the diversity of human genomes over repetitive regions can complicate the analysis process. It is crucial to ensure that the parameters chosen for visualization and subsequent analysis are appropriate for the specific genomic characteristics being examined. It may require careful consideration and expertise of a user to fine-tune the parameters and algorithms for optimization. Each main building unit (the vertex) of the MAP-graph represents a set of closely related sequence fragments. This is more analogous to the stringomics method proposed by Ferragina²¹ than other methods building graphs on top of MSA or variant calls. Such approaches combined with the sparse minimizers are efficient to reduce the computational complexity (fewer vertices) to represent larger-scale structures. Complementary to

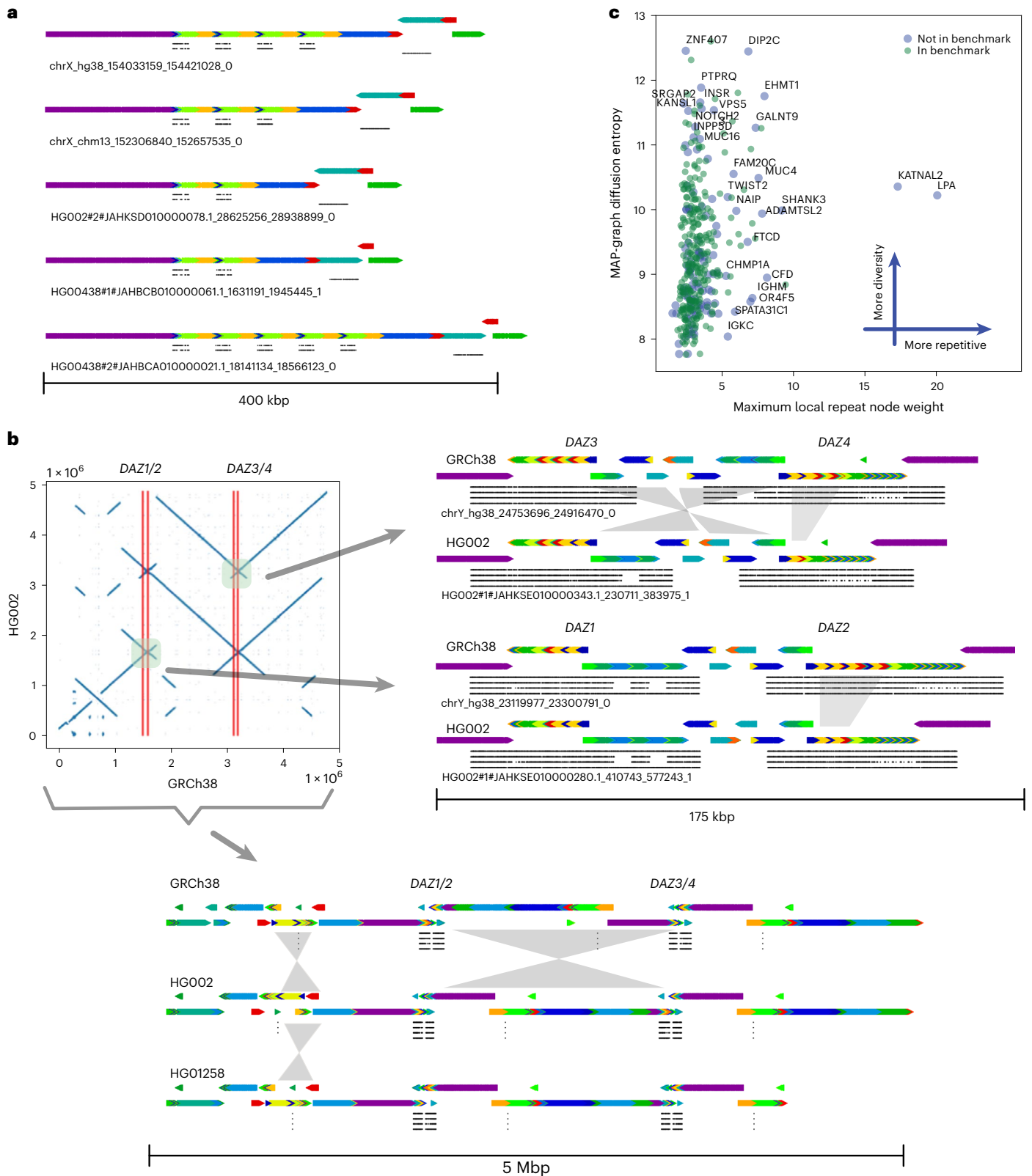


Fig. 4 | Principal bundle decomposition for genes in the repetitive regions of chromosome X and Y. a, MAP-graph principal bundle decomposition shows the repeat number changes of the *OPNILW*, *OPNIMW1/2/3* to *FLNA* loci. Auxiliary tracks are as follows: top *OPNILW*, middle *OPNIMW1/2/3* and bottom *FLNA*. **b**, The upper left image displays a dot plot comparing the HG002 assembly to GRCh38 over a 5 Mb region containing the *DAZ1/2/3/4* loci, highlighting an inversion between *DAZ1/2* and *DAZ3/4*. The image on the right provides a detailed view of the rearrangements at the gene scale level, with four tracks indicating the local matches to *DAZ1/2/3/4* from top to bottom. Comparison to

GRCh38's *DAZ2* reveals that the HG002 assembly is missing a segment (roughly 10 kb) of the darker green. The intergenic region between *DAZ3* and *DAZ4* also displays a rearrangement that can be described as an incomplete inversion or separate insertions and deletions. The bottom image shows a rearrangement at the whole locus, including all *DAZ1/2/3/4* over a 5 Mb region. The principal bundle decomposition reveals the different inverted structure of the HG002 T2T assembly and the HG1258 assembly compared to GRCh38. **c**, MAP-graph diffusion entropy versus repetitiveness survey for the 385 GIAB challenge CMRGs.

that, PGR-TK provides an interface to fetch the sequences within each vertex such that it is possible to combine a MAP-graph with other graph analysis approaches. For example, it can be integrated with Cactus graph²⁵ and A-de Bruijn Graph⁴² for base-level analysis, such as variant calling, genotyping and point mutation analysis, with recursive hybrid graph data structures.

We demonstrate how to use the PGR-TK for studying and characterizing the repetitive region *AMY1A* and the highly polymorphic HLA class II region. We present a tool in PGR-TK backed by a new pangenome graph traversal algorithm, relinearizing tangled graphs caused by repetitive sequences to principal bundles for visualization. With the principal bundle decomposition, we can automatically visualize the repetitive and non-repetitive components of haplotype assembly contig. The PGR-TK can provide intuitive qualitative information about different genome arrangement architectures with decomposition and associated visualizations. For example, it enables visualization of both the very large inversion in the *DAZ* locus and much smaller complex structural variation within the genes. The visualization of the *OPN1LW/OPN1MW* gene array shows subtle copy number variations, which can affect vision, as well as nearby structural variants. We also use PGR-TK to survey a set of regions of interest across the whole genome. We derive two metrics for measuring the polymorphism and repetitiveness in human pangenome to more systematically survey complexity of a large set of medically and clinically relevant genes. In the future, we aim to extend the PGR-TK library to provide more quantitative and base-level analysis for both fundamental and translational research using pangenome resources.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41592-023-01914-y>.

References

- Lander, E. S. et al. Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).
- Venter, J. C. et al. The sequence of the human genome. *Science* **291**, 1304–1351 (2001).
- Nurk, S. et al. The complete sequence of a human genome. *Science* **376**, 44–53 (2022).
- Siva, N. 1000 Genomes project. *Nat. Biotechnol.* **26**, 256 (2008).
- Canela-Xandri, O., Rawlik, K. & Tenesa, A. An atlas of genetic associations in UK Biobank. *Nat. Genet.* **50**, 1593–1599 (2018).
- Bycroft, C. et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature* **562**, 203–209 (2018).
- The ‘All of Us’ Research Program Investigators. The ‘All of Us’ Research Program. *N. Engl. J. Med.* **381**, 668–676 (2019).
- Voelkerding, K. V., Dames, S. A. & Durtschi, J. D. Next-generation sequencing: from basic research to diagnostics. *Clin. Chem.* **55**, 641–658 (2009).
- Rehder, C. et al. Next-generation sequencing for constitutional variants in the clinical laboratory, 2021 revision: a technical standard of the American College of Medical Genetics and Genomics (ACMG). *Genet. Med.* **23**, 1399–1415 (2021).
- Yohe, S. & Thyagarajan, B. Review of clinical next-generation sequencing. *Arch. Pathol. Lab. Med.* **141**, 1544–1557 (2017).
- Green, E. D., Rubin, E. M. & Olson, M. V. The future of DNA sequencing. *Nature* **550**, 179–181 (2017).
- Mahmoud, M. et al. Structural variant calling: the long and the short of it. *Genome Biol.* **20**, 246 (2019).
- Sedlazeck, F. J., Lee, H., Darby, C. A. & Schatz, M. C. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nat. Rev. Genet.* **19**, 329–346 (2018).
- De Coster, W., Weissensteiner, M. H. & Sedlazeck, F. J. Towards population-scale long-read sequencing. *Nat. Rev. Genet.* **22**, 572–587 (2021).
- Olson, N. D. et al. PrecisionFDA Truth Challenge V2: calling variants from short and long reads in difficult-to-map regions. *Cell Genom.* **2**, 100129 (2022).
- Aganezov, S. et al. A complete reference genome improves analysis of human genetic variation. *Science* **376**, eabl3533 (2022).
- Wang, T. et al. The Human Pangenome Project: a global resource to map genomic diversity. *Nature* **604**, 437–446 (2022).
- Sir, n, J. et al. Pangenomics enables genotyping of known structural variants in 5202 diverse genomes. *Science* **374**, abg8871 (2021).
- Liao, W.-W. et al. A draft human pangenome reference. *Nature* **617**, 312–324 (2023).
- Jarvis, E. D. et al. Semi-automated assembly of high-quality diploid human reference genomes. *Nature* **611**, 519–531 (2022).
- Ferragina, P. & Mishra, B. Algorithms in stringomics (I): pattern-matching against ‘stringomes’. Preprint at *bioRxiv* <https://doi.org/10.1101/001669> (2014).
- Garrison, E. & Guarracino, A. Unbiased pangenome graphs. *Bioinformatics* **39**, btac743 (2023).
- Eizenga, J. M. et al. Pangenome graphs. *Annu. Rev. Genom. Hum. Genet.* **21**, 139–162 (2020).
- Li, H., Feng, X. & Chu, C. The design and construction of reference pangenome graphs with minigraph. *Genome Biol.* **21**, 265 (2020).
- Paten, B. et al. Cactus graphs for genome comparisons. *J. Comput. Biol.* **18**, 469–481 (2011).
- Ebler, J. et al. Pangenome-based genome inference allows efficient and accurate genotyping across a wide spectrum of variant classes. *Nat. Genet.* **54**, 518–525 (2022).
- Garrison, E. et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* **36**, 875–879 (2018).
- Computational Pan-Genomics Consortium. Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* **19**, 118–135 (2018).
- Hickey, G. et al. Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome Biol.* **21**, 35 (2020).
- Armstrong, J. et al. Progressive Cactus is a multiple-genome aligner for the thousand-genome era. *Nature* **587**, 246–251 (2020).
- Beller, T. & Ohlebusch, E. A representation of a compressed de Bruijn graph for pan-genome analysis that enables search. *Algorithms Mol. Biol.* **11**, 20 (2016).
- Darling, A. C. E., Mau, B., Blattner, F. R. & Perna, N. T. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res.* **14**, 1394–1403 (2004).
- Guarracino, A., Heumos, S., Nahnsen, S., Prins, P. & Garrison, E. ODGI: understanding pangenome graphs. *Bioinformatics* **38**, 3319–3326 (2022).
- Chin, C.-S. et al. A diploid assembly-based benchmark for variants in the major histocompatibility complex. *Nat. Commun.* **11**, 4794 (2020).
- Wagner, J. et al. Curated variation benchmarks for challenging medically relevant autosomal genes. *Nat. Biotechnol.* **40**, 672–680 (2022).
- Bhowmick, B. K., Satta, Y. & Takahata, N. The origin and evolution of human ampliconic gene families and ampliconic structure. *Genome Res.* **17**, 441–450 (2007).
- Zook, J. M. et al. A robust benchmark for detection of germline large deletions and insertions. *Nat. Biotechnol.* **38**, 1347–1355 (2020).
- Cody, S. Year 1 Sequencing data release. *Human Pangenome Reference Consortium* <https://humanpangenome.org/year-1-sequencing-data-release/> (2021).

39. Chin, C.-S. & Khalak, A. Human genome assembly in 100 minutes. Preprint at *bioRxiv* <https://doi.org/10.1101/705616> (2019).
 40. Ekim, B., Berger, B. & Chikhi, R. Minimizer-space de Bruijn graphs: whole-genome assembly of long reads in minutes on a personal computer. *Cell Syst.* **12**, 958–968.e6 (2021).
 41. Li, H. Minimap and minimiasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* **32**, 2103–2110 (2016).
 42. Pevzner, P. A., Tang, H. & Tesler, G. De novo repeat classification and fragment assembly. *Genome Res.* **14**, 1786–1796 (2004).
 43. Jiang, Z. et al. Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nat. Genet.* **39**, 1361–1368 (2007).
 44. Wang, M., Ye, Y. & Tang, H. A de Bruijn graph approach to the quantification of closely-related genomes in a microbial community. *J. Computat. Biol.* **19**, 814–825 (2012).
 45. Pu, L., Lin, Y. & Pevzner, P. A. Detection and analysis of ancient segmental duplications in mammalian genomes. *Genome Res.* **28**, 901–909 (2018).
 46. Deorowicz, S., Danek, A. & Li, H. AGC: compact representation of assembled genomes. *Bioinformatics* **39**, btad097 (2023).
 47. Project jupyter (Jupyter, 2023); <https://jupyter.org>
 48. Roberts, M., Hayes, W., Hunt, B. R., Mount, S. M. & Yorke, J. A. Reducing storage requirements for biological sequence comparison. *Bioinformatics* **20**, 3363–3369 (2004).
 49. Ondov, B. D. et al. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.* **17**, 132 (2016).
 50. Gibbs, A. J. & McIntyre, G. A. The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem.* **16**, 1–11 (1970).
 51. Kececioğlu, J. D. & Myers, E. W. Combinatorial algorithms for DNA sequence assembly. *Algorithmica* **13**, 7 (1995).
 52. Myers, E. W. The fragment assembly string graph. *Bioinformatics* **21**, ii79–ii85 (2005).
 53. Trowsdale, J. & Knight, J. C. Major Histocompatibility Complex genomics and human disease. *Annu. Rev. Genomics Hum. Genet.* **14**, 301–323 (2013).
 54. Vandiedonck, C. & Knight, J. C. The human Major Histocompatibility Complex as a paradigm in genomics research. *Brief. Funct. Genomics Proteom.* **8**, 379–394 (2009).
 55. Horton, R. et al. Variation analysis and gene annotation of eight MHC haplotypes: the MHC Haplotype Project. *Immunogenetics* **60**, 1–18 (2008).
 56. Gardner, J. C., Michaelides, M. & Hardcastle, A. J. Cone opsins, colour blindness and cone dystrophy: genotype-phenotype correlations. *S. Afr. Med. J.* **106**, S75–S78 (2016).
 57. Moro, E. et al. Male infertility caused by a de novo partial deletion of the DAZ cluster on the Y Chromosome1. *J. Clin. Endocrinol. Metab.* **85**, 4069–4073 (2000).
 58. lh3/dipcall. *GitHub* <https://github.com/lh3/dipcall> (2021).
 59. Gardner, J. C. et al. X-linked cone dystrophy caused by mutation of the red and green cone opsins. *Am. J. Hum. Genet.* **87**, 26–39 (2010).
- Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
- Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.
- © The Author(s) 2023

Methods

Sequence database and SHIMMER index construction

To generate the SHIMMER index, each sequence was scanned and the symmetrical minimizers were generated with the specific minimizer window size w and kmer size k . We called this the first level of minimizer. Given a reduction factor $r > 1$, additional levels of minimizer sets were generated to increase the span between the minimizers by a reduction step to facilitate pangenomics analysis. Even with the reduction step, in some simple sequence context, for example, long or short tandem repeats, two minimizers can remain too close to each other. A parameter 'min_span' can be applied to eliminate a pair of minimizers that are too close. We used a heuristic algorithm to eliminate those minimizers that were within the distance of min_span to each other. This helped to reduce the minimizer density when detailed analysis for those simple context regions was not desired. Setting min_span to zero and r to one generated the standard minimizers for each sequence.

Each pair of the reduced minimizers (SHIMMERs) was used as the key to build a hashmap to the sequence ID, and coordinates and matching orientation of the minimizer pairs on the sequences.

We also present examples of API calling and command line usage below.

Generate MAP-graph

The MAP-graph is constructed by scanning through each sequence in the database. The vertices are simply the set of the tuples of neighboring minimizers (minimizer anchored segments). The edges are constructed by connecting minimizer anchored segments as a bidirected graph. One can consider this to be an extension of the string graph⁵² where the overlaps are the minimizers at both ends. However, in the pangenome graph, each vertex includes a set of sequence segments from multiple genomes rather than one sequence.

As the MAP-graph can be constructed by scanning the SHIMMER pairs through the sequences. For a given set of n sequences, $S = \{s_i | i = 0 \dots n - 1\}$, the vertices of the MAP-graph are

$$V = \{ (m^{(i)}_p, m^{(i)}_{p+1}) | m^{(i)}_p \text{ and } m^{(i)}_{p+1} \text{ are the } p\text{th and } (p+1)\text{th} \\ \text{minimizers of a sequence } s_i, \text{ in all } s_i \text{ in } S \}.$$

We can assign a weight w_i of a vertex $v_i = (m_a, m_b)$ as the total number of observed (m_a, m_b) pairs in S .

The edges of the MAP-graph are

$$E = \{ (v_i, v_j) | v_i = (m^{(i)}_p, m^{(i)}_{p+1}) \text{ and } v_j = (m^{(i)}_{p+1}, m^{(i)}_{p+2}) \text{ for all} \\ (m^{(i)}_p, m^{(i)}_{p+1}, m^{(i)}_{p+2}) \text{ in all } s_i \text{ in } S \}$$

Identify the principal bundles in a MAP-graph

To decompose a MAP-graph into principal bundles for downstream analysis, we apply a variation of depth first search⁶⁰ (DFS) to build the traversal trees from the graph. Our DFS prioritized vertices with high 'weight' (defined as the number of sequence segments contained in a vertex) and accounting for the bidirected nature of the MAP-graph.

The DFS traversal through the graph is then converted to a tree structure internally. The leaf nodes in the tree are typically when the DFSs are terminated by no out edge from a node or a bubble or a loop is found. As we prioritize the weights of the vertices during DFS, long paths usually correspond to the 'common' paths that most sequences in the data would go through. Rarer haplotypes typically correspond to short bubble paths in the MAP-graph. Thus, they can be identified as short branches in the DFS traversal tree. We use the tree to remove those vertices in the MAP-graph if those are shorter than prespecified length in the DFS tree. In general, the vertices in the principal bundle represent more sequences in the set of input sequences (Supplementary Fig. 6) and are likely more conservative in the pangenome.

After removing the vertices corresponding to the short branches, we further remove vertices in the MAP-graph that have more than three out edges after converting the MAP-graph as an undirected graph. After such removal, the graph will only consist of simple paths, and we output those paths as the principal bundles.

In summary, here is the sketch of the algorithm:

- (1) Build a DFS traversal tree with a deep first search for a given MAP-graph. To capture paths that are more conserved among the pangenome sequences of interest as the principal bundles, our DFS search prioritizes high-weight vertices when constructing the DFS traversal tree
- (2) In MAP-graph, remove vertices that correspond to nodes in short branches of the DFS traversal tree
- (3) Remove branching vertices in MAP-graph (by considering it as an undirected graph)
- (4) Output the simple paths from the resulting graph as the principal bundles.

Principal component plot for the HLA class II locus

To generate the principal component of the pangenome HLA class II sequences, we convert each of the haplotype sequences to a binary vector. The binary vector has the same length of the total number of vertices of all principal bundles. Let us call these vertices $V = \{v_i | v_i$ in principle bundles, $i = 0 \dots n-1\}$, where n is the total number of vertices in the principal bundles. For each sequence s , we construct a binary vector $w_s = \{b_0, b_1, \dots, b_{n-1}\}$ where $b_i = 1$ if the sequence s contains the vertex v_i , and $b_i = 0$ if not. Then, we perform the standard principal component transformation with the binary vectors of all sequences from the HLA class II region.

General workflow for analyzing a region of interest

Here we outline the general workflow on how to use PGR-TK to generate MAP-graph and the principal bundle decomposition.

- (1) For a region or sequences of interest, put the sequences as a fasta file for querying the PGR-TK pangenome sequence database. (PGR-TK provides a command line tool 'pgr-fetch' and python APIs to fetch such sequence from the PGR-TK sequence database.)
- (2) Query the whole pangenome database to get initial hits that match the query sequence with the command line tool 'pgr-query' or using the Python APIs.
- (3) Filter the hits to remove unwanted matches that do not match a user's analysis objectives. With the command line tool 'pgr-query', it generates a summary table of the hits for filtering.
- (4) With the filtered sequences, using 'pgr-pbundle-decomp' command line tool to generate the MAP-graph in GFA format and principal bundle decomposition in the BED format. Beside analyzing the generated data, the generated bed file of the decomposition can be rendered by the pgr-pbundle-bed2svg to generate visualization. Python APIs are also provided for more scripting to resolve complicated analysis cases.
- (5) Optionally, we can take the fetched sequences from 'pgr-query' to use with other third party tools, for example, calling variant with dipcall, create an MSA, or building other local pangenome graphs with minigraph or pggg.
- (6) Re-adjust the parameters ($w, k, r, \text{min_span}$) and repeat (3) and (4) with additional analysis of the results if necessary.

Building index

For a large sequence set, for example 47 whole-genome HPRC assemblies, PGR-TK uses the AGC format⁴⁶ to store the sequence efficiently. A command line tool 'pgr-mdb' is developed with the PGR-TK package to create the index file on top of the AGC file. For example, for a prebuild HPRC year one assembly AGC file (1.33 Gb), we create a file

(/data/pgr-tk-HGRP-y1-evaluation-set-v0_input) include a file system path to the AGC file, /data/gr-tk-HGRP-y1-evaluation-set-v0.agc and call 'pgr-mdb' to create the index files with a prespecified prefix (/data/pgr-tk-HGRP-y1-evaluation-set-v0):

```
echo /data/pgr-tk-HGRP-y1-evaluation-set-v0.agc \  
> /data/pgr-tk-HGRP-y1-evaluation-set-v0_input  
/code/pgr-mdb /data/pgr-tk-HGRP-  
y1-evaluation-set-v0_input \  
/data/pgr-tk-HGRP-y1-evaluation-set-v0
```

Two files will be generated in this example

```
/data/pgr-tk-HGRP-y1-evaluation-set-v0.mdb # 15 Gb  
for (w, k, r, min_span) = (80,56,4,64)  
/data/pgr-tk-HGRP-y1-evaluation-set-v0.midx # 3.1 Mb
```

The index and sequence data can be loaded into a python workspace by

```
import pgrtk  
sdb = pgrtk.SeqIndexDB()  
sdb.load_from_agc_index('pgr-tk-HGRP-y1-evaluation-  
set-v0')
```

As the indexes are loaded into memory, we suggest using a computing instance that has a random access memory larger than about four times that of the index file to avoid swapping thrashing.

For smaller sequence files, the sequence database object (for example, the 'sdb' in the example above) created by pgrtk.SeqIndexDB() can create and load sequences using load_from_fastx() method. See the library documentation at <https://genedx.github.io/pgr-tk/> for more detailed descriptions of all python objects, methods and functions in the PGR-TK package.

Query sequence in the PGR-TK sequence database

Command line example. A command line tool named pgr-query is provided to query a PGR-TK database with a set of sequences, each of which represents a region of interest. It is recommended to select regions larger than 20 kb, as they contain enough SHIMMER anchors. If smaller regions are of interest, padding with flanking sequences can improve the results.

The follow command shows an example querying the database: pgr-query/data/pgr-tk-HGRP-y1-evaluation-set-v0 ROI_seq.fa pg_seqs --merge-range-tol 100,000.

In this example, the ROI_seq.fa file contains sequences from the regions of interest. The pgr-query tool generates a set of fasta files with the prefix 'pg_seqs' for each query sequence in the 'ROI_seq.fa' file. Additionally, a 'pg_seqs.hit' file is produced, which contains information about the alignment range between the query and the results, as well as the number of anchors identified between each pair of query results. This information can be used to filter out unwanted alignments.

Python API example. For finding homologous sequences in a PGR-TK database, we need to start with a query sequence. We can fetch a sequence in the database giving a known 'data source', 'contig name' tuple and the beginning and ending coordinates. As the SHIMMERS are sparsely distributed in a sequence, the query sequence should be long enough to cover enough minimizer anchors. The python statement shows a typical code fragment to generate query results of a region of interest:

```
ref_file_name, roi_chr, roi_b, roi_e = 'hg19_  
tagged.fa', 'chr6_hg19', 32130918, 32959917  
padding = 10000
```

```
#get a segment of a reference  
roi_seq = ref_db.get_sub_seq(ref_file_name, roi_  
chr, roi_b-padding, roi_e+padding)  
# using the roi_seq to find hits in 'sdb'  
aln_range = pgrtk.query_sdb(sdb, roi_seq,  
merge_range_tol=200000)
```

The output aln_range from the query_sdb() call contains data of the hits in the PGR-TK database. Internally, the query_sdb() method performs:

- (1) create SHIMMER pairs of the query sequence (2) use the SHIMMER pairs and the hashmap index to find all hits in the database
- (2) perform sparse dynamic programming to find sparse alignments between the query sequence and all hits in the database
- (3) merge the alignment segments if any of them are within the merge_range_tol parameter.

The parameter merge_range_tol is introduced to avoid alignment fragmentation when the query sequence contains a region of high polymorphism but we still want to fetch those diverse sequences for constructing the pangenome graph.

Typically, a user needs to process the data in aln_range for different analysis. Our example Jupyter Notebooks provides various examples for processing the output to generate dot plot or MAP-graphs and so on.

Build minimizer anchor pangenome graph and principal bundle decomposition

Command line example. Given a set of sequence in a fasta file, for example, the query results from pgr-query command, we can build the pangenome graph and the principal bundle decomposition (outputted as a bed file) by

```
pgr-pbundle-decomp -w 48 -k 56 -r 8 \  
--min-span 12 --bundle-length-cutoff 100 \  
--bundle-merge-distance 1000 --min-branch-size 8 \  
--min-cov 0 --include file_contain_contig_names \  
pgr-query pgr_out
```

In the pangenome graph construction process, the following options can be used to control the graph construction: -w, -k, -r and --min-span. The --min-cov option sets the minimum coverage requirement for a vertex to be included in the principal bundle graph. The --min-branch-size option allows the user to filter out short branches that contain less than a specified number of vertices in the MAP-graph. The --bundle-length-cutoff option allows the user to exclude bundles shorter than the specified length. When two bundles have the same identifier and are within the distance specified by --bundle-merge-distance, they will be merged. If the --include option is specified, only those contigs specified in the file_contain_contig_names file will be analyzed.

The command generates a set of files with the prefix pgr_out:

```
pgr_out.bed # the bed file contains the principal  
bundle decomposition  
pgr_out.ctg.summary.tsv # summary for bundle  
statistics for each contig  
pgr_out.mapg.gfa # MAP-graph in GFA format  
pgr_out.mapg.idx # index for the sequence in the  
GFA file  
pgr_out.pmapg.gfa #principal bundle graph in the  
GFA format
```

The pgr_out.bed has the following format. Each line is a bundle contain in a contig: contig_name begin_coordinate end_coordinate bundle_specification where the bundle_specification is six fields delimited by ':'. The fields are bundle identifier, bundle vertex count,

orientation, begin vertex number, end vertex number, 'R' or 'U' for repetitive or unique in the contig. Note such a region in a contig may not only project to the full bundle. In such cases, `end_vertex-bgn_vertex` is less than `bundle_vertex_count`.

The `*.mapg.idx` file contains the information about each vertex in the `.gfa` files. It contains three kinds of tagged line starting with 'K', 'C' and 'F'. The K line specifies the parameters used to generate the graph. The C lines specify the contigs contained in the graph. The F lines specify the fragment of sequences contained in each vertex. Here are the fields for echo of them: K line: *w, k, r, min_span*

S line: *contig_idenfiter, contg_name, contig_source, contig_length*

F line: *frag_unique_idenfiter, frag_numeric_idenfiter, contig_idenfiter, start_coordinate, end_coordinate, orientation*

Python API:

To build the MAP-graph and principal bundle decomposition within a python program, one can use the instance methods `generate_mapg_gfa()` and `get_principal_bundle_decomposition()` of an `pgrtk.SeqIndexDB()` object. Please read the documentation at <https://genedx.github.io/pgr-tk/> for the API details.

Clustering principal bundle decomposition structure

The follow command performs clustering of the principal bundles stored in a bed file: `pgr-pbundle-bed2dist pgr_out.bed pgr_out`.

It will generate these output files: `pgr_out.dist` # all pairwise distance

```
pgr_out.nwk # a hierarchical tree in Newick format
pgr_out.ddg # a file contain the dendrogram for the
pgr-pbundle-bed2svg to draw the dendrogram panel
```

Generate principal bundle decomposition plot in scalable vector graph format

The following is an example to generate a SVG file `pgr_out.svg` from `pgr_out.bed` that layout 500,000 bp with annotation specified in a file called `pgr_annotation` with a dendrogram panel on the left from the `pgr_out.ddg` dendrogram data

```
/code/pgr-pbundle-bed2svg pgr_out.bed pgr_out \
--track-range 500000 --track-tick-interval 100000 \
--track-panel-width 1200 --stroke-width 0.5 \
--annotations pgr_annotation \
--ddg-file pgr_out.ddg \
--highlight-repeats 3
```

Survey on the genome in bottle challenging clinical and medically relevant genes with the MAP-graphs

The GIAB Consortium provides variant call benchmarks on seven benchmark genomes^{35,61,62}. These benchmarks were initially formed by integrating multiple short-read technologies, but the latest version integrated linked-read and long-read technologies to form benchmarks in regions difficult to map with short reads. However, a set of 395 challenging but medically relevant genes were identified as substantially (more than 10%) excluded from the mapping-based benchmark due to long repeats, large structural variations, segmental duplications and/or high polymorphism between the benchmark genome HG002 and the reference genome hg19 or GRCh38. A long-read genome assembly approach provided a reliable benchmark call set of 273 out of the 395 genes³⁵. The remaining 122 were still excluded mainly because they were not accurately assembled, or no benchmarking tools exist to compare different representations of complex variants in the genes. Overall, 395 genes have recently shown to include high levels of polymorphism across different ethnicities making it highly challenging to represent their variations⁶³.

The current GIAB variant benchmarks focus on a small set of seven well-characterized genomes with extensive short-, linked- and long-read data to ensure robust benchmarks and more tractable

method development and benchmark evaluations. Meanwhile, a limited representation of genomes in a population may miss significant structural variants, additional copies of genes and context for important variants for diseases not observed in a smaller dataset. Given that increasingly accurate long-read and assembly level data are being produced at pangenome scales now, we are surveying how we can use such resources to benchmark variant accuracy at a broader population level for the challenging CMRG. Such pangenome analysis will help to generate guidelines for future practice.

With PGR-TK, we extract all sequences from the HPRC year one release (94 haplotype assemblies), and CHM13 v.1.1, GRCh38 and hg19 of all 385 CMRG. We generate a MAP-graph of each gene and output this in GFA format. For each graph, we derived two metrics to estimate (1) the degree of polymorphism among the pangenomes, and (2) the repeat content taking account of the variations of the pangenomes.

To estimate the degree of polymorphism, we consider a diffusion process in the graph (below) and derive an entropy-like quantity from a normalized equilibrium distribution. The higher entropy values indicate more complicated graphs likely from the polymorphism from different genomes. The diffusion weight on each vertex is also associated with multiplicity and repetitiveness of the corresponding segments in the pangenomes. We pick the average of the top 32 diffusion weights from the MAP-graph of each gene as a simple metric to measure the most challenging repeat content within a gene.

To gain insight about the challenge for calling variants of the CMRG set at a pangenome scale, we plot the diffusion entropy versus the maximum local repeat weights for each gene (Fig. 4c). As there are no obvious correlations, these two quantities provide independent measurements of two aspects of the MAP-graph structures of these genes. We find high repetitive genes are harder to create a reliable variant benchmark call set for. Many highly repetitive genes are excluded from the current CMRG benchmark set. We do not observe that higher entropy is correlated with the reduction of the gene in the benchmark set.

We highlight several genes with high entropy or high repetitiveness. The MAP-graphs and the Integrative Genomics Viewer view of the pangenome assemblies of a selected set genes (*LMF1*, *ANKRD11*, *SRGAP2*, *KMT2C*, *LPA*, *MUC4*, *MUC3A*, *KATNAL2*, *FLG*) aligned to GRCh38 are shown in Supplementary Fig. 4. In the Integrative Genomics Viewer view of *LMF1* (Supplementary Fig. 4a), several variation hotspots are visible and may correspond to localized structural variants. This serves as a simple example to examine the concordance of structural variation in the population and the principal bundle decomposition. We use PAV⁶⁴ to call structural variants for comparison. In Supplementary Fig. 4b, the structural variant calls resulting from PAV are provided as an auxiliary track (black) below the principal bundle decomposition tracks for a selected set of HPRC year one genomes, where both haplotypes are resolved in the region. It is evident that the structural variation calls correspond to the regions where principal bundles have complex structure and are distinct from the reference genome GRCh38 (the top track). As shown in *LPA* (associated with coronary disease⁶⁵, Supplementary Fig. 7), *KATNAL2* (loss of function variant discovered in autistic proband^{66,67}) also has long tandem repeat variations in the HPRC pangenome cohort. We find the number of the 5.8 kbp repeats inside *KATNAL2* ranges from 3 to 25 (Supplementary Fig. 8). Applying MAP-graph decomposition on genes such as *KATNAL2* with a big pangenome reference panel will provide additional insights to the natural of the variability of the repetitiveness and its effects on the underlying biology like other, more well-studied genes, for example, *LPA*, in the coming years.

Compute graph diffusion entropy and max repetitive weight

It would be desirable to derive quantitative measurements so we can characterize a set of large numbers of MAP-graphs fast. One thing we are interested in quantifying is how complex a graph is. The intuition is that if a region of the genome is more polymorphic in the population, the graph will have more alternative paths, or bubbles. We like

to generate a quantity as a proxy for that. For this, we borrow the idea from network science study and spectral graph theory to consider a diffusion/random walk process on a graph⁶⁸. For a graph, we consider a set of random walkers starting at each vertex. The random walkers can drift on the graph through the edge-connection. We can consider the distribution of the random walkers in the final equilibrium state. If a graph is relatively simple, then the final distribution will be uniform (subject to minor boundary condition corrections). On the other hand, if the weights of the vertices or topology of the graph are more complex, we would expect the final distribution of the walkers would be less uniform and reflect the complicated nature of the graph.

The final distribution of the such diffusion process can be obtained by simple matrix multiplication iteration from the adjacent matrix of a MAP-graph. Given an adjacency matrix A , where the matrix element A_{ij} is the number of sequences supports edge from v_i to v_j . The final distribution P can be written as $P = (1/N) \lim_{n \rightarrow \infty} M^n \mathbf{1}$, where $M = AD^{-1}$, D is the degree matrix defined as D_{ii} being the degree of vertex v_i and $D_{ij} = 0$ if $i \neq j$, N is the total number of vertices and $\mathbf{1}$ is a column vector in which every element is one. (When we compute P , we only repeat the number of multiplications N times to approximate the final distribution.)

\mathbf{P} is a normalized column vector $[p_0, p_1, \dots, p_{n-1}]^T$ such that $\sum_{i=0, \dots, n-1} p_i = 1$ (see Supplementary Fig. 4a for an example). The diffusion entropy used in this work is defined as $S = -\sum_{i=0, \dots, n-1} p_i \log_2(p_i)$.

To find the highly repetitive elements inside a region of interest, we look into largest elements in the unnormalized vector \mathbf{NP} as a proxy of average number of repeats considering the graph structure. We pick the top 32 elements in \mathbf{NP} and use the average of those as a proxy number to estimate the repetitiveness of potential repeat units inside a region of interest.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The HPRC year one release sequence and prebuilt index: <https://giab-data.s3.amazonaws.com/PGR-TK-Files/pgr-tk-HGRP-y1-evaluation-set-v0.tar> and https://figshare.com/articles/journal_contribution/HPRC-Y1-sequence-data_zip/22584403 (<https://doi.org/10.6084/m9.figshare.22584403>). Scripts and source data URLs for constructing the HPRC AGC file: <https://github.com/GeneDX/pgr-tk-notebooks/tree/main/pgr-tk-sequence-source>. All GFA files, fetched sequences from HPRC year one release of the 385CMRG: https://giab-data.s3.amazonaws.com/PGR-TK-Files/CMRG_output_dir_v0.3.3.tar.

Code availability

The code of the PGR-TK is hosted in a GitHub repository, <https://github.com/GeneDX/pgr-tk>, and the code can be obtained using the standard 'git clone' command. Follow the README.md document for building the PGR-TK. A document for Python API is provided at <https://genedx.github.io/pgr-tk/>. The source code for the v.0.4.1 version is also hosted at https://figshare.com/articles/journal_contribution/pgr-tk-0_4_1_zip/22584793 (<https://doi.org/10.6084/m9.figshare.22584793>). Prebuilt binaries for Linux Ubuntu distribution 22.04 can be downloaded from <https://github.com/GeneDX/pgr-tk/releases/tag/v0.4.1> and https://figshare.com/articles/journal_contribution/pgr-bin-pacakege-v0_4_1_tgz/22584787 (<https://doi.org/10.6084/m9.figshare.22584787>). Example Notebooks using PGR-TK including code making most of (the source of) the plots in this paper can be found at <https://github.com/GeneDX/pgr-tk-notebooks> and https://figshare.com/articles/journal_contribution/scripts_and_artifacts_zip/22584118 (<https://doi.org/10.6084/m9.figshare.22584118>). Additional information about a docker image with prebuilt PGR-TK library and Jupyter Laboratory Server and usage is at <https://github.com/GeneDX/pgr-tk/blob/main/pgr-tk-workstation/Readme.md>.

References

60. Skiena, S. S. *The Algorithm Design Manual* (Springer, 2008).
61. Zook, J. M. & Salit, M. Advancing benchmarks for genome sequencing. *Cell Syst.* **1**, 176–177 (2015).
62. Wagner, J. et al. Benchmarking challenging small variants with linked and long reads. *Cell Genom.* **2**, 100128 (2022).
63. Ji, Y., Gong, J., Sedlazeck, F. J. & Fan, S. Characterizing the genetic polymorphisms in 370 challenging medically relevant genes using long-read sequencing data from 41 human individuals among 19 global populations. Preprint at *bioRxiv* <https://doi.org/10.1101/2022.08.03.502734> (2022).
64. Ebert, P. et al. Haplotype-resolved diverse human genomes and integrated analysis of structural variation. *Science* **372**, eabf7117 (2021).
65. Burgess, S. et al. Association of LPA variants with risk of coronary disease and the implications for lipoprotein(a)-lowering therapies: a Mendelian randomization analysis. *J. Am. Med. Assoc. Cardiol.* **3**, 619–627 (2018).
66. Sanders, S. J. et al. De novo mutations revealed by whole-exome sequencing are strongly associated with autism. *Nature* **485**, 237–241 (2012).
67. O’Roak, B. J. et al. Sporadic autism exomes reveal a highly interconnected protein network of de novo mutations. *Nature* **485**, 246–250 (2012).
68. Spielman, D. A. Spectral graph theory and its applications. in *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)* 29–38 (IEEE, 2007).

Acknowledgements

We thank B. Mishra for reading and commenting on the draft of the manuscript. Certain commercial equipment, instruments or materials are identified to specify adequately experimental conditions or reported results. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment, instruments or materials identified are necessarily the best available for the purpose. F.J.S. and S.B. are supported by National Institutes of Health grants (nos. UM1HG008898 and 1U01HG011758-01). P.S. is supported by an National Institutes of Health/ National Institute of General Medical Sciences grant (no. R35GM142916).

Author contributions

Conceptualization and design were done by C.-S.C., A.K., F.J.S., P.H.S., J.W. and J.M.Z. Algorithm, code and documentation development were done using PGR-TK by C.-S.C. The manuscript was written by C.-S.C., A.K., F.J.S. and J.M.Z. Manuscript revision, data management and code validation were carried out by S.B., F.J.S., P.H.S. and J.W.

Competing interests

C.-S.C. is an employee and shareholder of GeneDX. F.J.S. obtains research support from Illumina, PacBio and Oxford Nanopore. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41592-023-01914-y>.

Correspondence and requests for materials should be addressed to Chen-Shan Chin.

Peer review information *Nature Methods* thanks Bjarni Halldorsson and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Lin Tang, in collaboration with the *Nature Methods* team.

Reprints and permissions information is available at www.nature.com/reprints.

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. F , t , r) with confidence intervals, effect sizes, degrees of freedom and P value noted
Give P values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's d , Pearson's r), indicating how they were calculated

Our web collection on [statistics for biologists](#) contains articles on many of the points above.

Software and code

Policy information about [availability of computer code](#)

Data collection

The scripts to processes the origin data are in <https://github.com/GeneDX/pgr-tk-notebooks/tree/main/pgr-tk-sequence-source> (link included in the manuscript).

software version:

AGC (Assembled Genomes Compressor) v. 2.0 [build 20220405.1]

PGR-TK: revision: 75fa20b41592941c9e6eef3f914d97788ee06b86 and after

Data analysis

The following third party software are used in analysis

minimap2:

source: <https://github.com/lh3/minimap2>

revision: 01b98e8e52a8acfed5a9d57853f028267eaf045f

seqwish:

source: <https://github.com/ekg/seqwish.gi>

revision: f362f6f5ea89dbb6a0072a8b8ba215e663301d33

minigraph

source: <https://github.com/lh3/minigraph>

revision: 3398263be225ba923140a1081b505b71f2cdf8fb

rustc: 1.68.2 (9eb3afe9e 2023-03-27)

agc: <https://github.com/cschin/agc.git> revision: 453c0afdc54b4aa00fa8e97a63f196931fdb81c4 (git repo forked from <https://github.com/>)

refresh-bio/agg)

PGR-TK: revision: 75fa20b41592941c9e6eef3f914d97788ee06b86 and after

Code repository:

source: <https://github.com/GeneDx/pgr-tk>API document: <https://genedx.github.io/pgr-tk/>

pre-built binaries for amd64 Ubuntu 20.04 linux distribution

<https://github.com/GeneDx/pgr-tk/releases/tag/v0.4.0>The licenses of the code / usage limitation: <https://github.com/GeneDx/pgr-tk/blob/main/LICENSE>Additional examples can be found at <https://github.com/GeneDx/pgr-tk> and <https://github.com/Sema4-Research/pgr-tk-notebooks>

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research [guidelines for submitting code & software](#) for further information.

Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The initial assembly contigs are downloaded from AWS S3, the full GenBank Accessions are in https://github.com/GeneDx/pgr-tk-notebooks/blob/main/pgr-tk-sequence-source/genbank_accessions.txt. The URLs that we download are in https://github.com/GeneDx/pgr-tk-notebooks/blob/main/pgr-tk-sequence-source/Year1_assemblies_v2_genbank.index, we list them as well:

s3://human-pangenomics/working/HPRC/HG00438/assemblies/year1_f1_assembly_v2_genbank/HG00438.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00438/assemblies/year1_f1_assembly_v2_genbank/HG00438.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00621/assemblies/year1_f1_assembly_v2_genbank/HG00621.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00621/assemblies/year1_f1_assembly_v2_genbank/HG00621.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00673/assemblies/year1_f1_assembly_v2_genbank/HG00673.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00673/assemblies/year1_f1_assembly_v2_genbank/HG00673.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00735/assemblies/year1_f1_assembly_v2_genbank/HG00735.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00735/assemblies/year1_f1_assembly_v2_genbank/HG00735.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00741/assemblies/year1_f1_assembly_v2_genbank/HG00741.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG00741/assemblies/year1_f1_assembly_v2_genbank/HG00741.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01071/assemblies/year1_f1_assembly_v2_genbank/HG01071.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01071/assemblies/year1_f1_assembly_v2_genbank/HG01071.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01106/assemblies/year1_f1_assembly_v2_genbank/HG01106.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01106/assemblies/year1_f1_assembly_v2_genbank/HG01106.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01123/assemblies/year1_f1_assembly_v2_genbank/HG01123.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01123/assemblies/year1_f1_assembly_v2_genbank/HG01123.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01175/assemblies/year1_f1_assembly_v2_genbank/HG01175.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01175/assemblies/year1_f1_assembly_v2_genbank/HG01175.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01258/assemblies/year1_f1_assembly_v2_genbank/HG01258.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01258/assemblies/year1_f1_assembly_v2_genbank/HG01258.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01358/assemblies/year1_f1_assembly_v2_genbank/HG01358.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01358/assemblies/year1_f1_assembly_v2_genbank/HG01358.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01361/assemblies/year1_f1_assembly_v2_genbank/HG01361.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01361/assemblies/year1_f1_assembly_v2_genbank/HG01361.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01891/assemblies/year1_f1_assembly_v2_genbank/HG01891.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01891/assemblies/year1_f1_assembly_v2_genbank/HG01891.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01928/assemblies/year1_f1_assembly_v2_genbank/HG01928.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01928/assemblies/year1_f1_assembly_v2_genbank/HG01928.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01952/assemblies/year1_f1_assembly_v2_genbank/HG01952.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01952/assemblies/year1_f1_assembly_v2_genbank/HG01952.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01978/assemblies/year1_f1_assembly_v2_genbank/HG01978.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG01978/assemblies/year1_f1_assembly_v2_genbank/HG01978.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02148/assemblies/year1_f1_assembly_v2_genbank/HG02148.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02148/assemblies/year1_f1_assembly_v2_genbank/HG02148.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02257/assemblies/year1_f1_assembly_v2_genbank/HG02257.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02257/assemblies/year1_f1_assembly_v2_genbank/HG02257.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02486/assemblies/year1_f1_assembly_v2_genbank/HG02486.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02486/assemblies/year1_f1_assembly_v2_genbank/HG02486.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02559/assemblies/year1_f1_assembly_v2_genbank/HG02559.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02559/assemblies/year1_f1_assembly_v2_genbank/HG02559.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02572/assemblies/year1_f1_assembly_v2_genbank/HG02572.paternal.f1_assembly_v2_genbank.fa.gz

s3://human-pangenomics/working/HPRC/HG02572/assemblies/year1_f1_assembly_v2_genbank/HG02572.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02622/assemblies/year1_f1_assembly_v2_genbank/HG02622.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02622/assemblies/year1_f1_assembly_v2_genbank/HG02622.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02630/assemblies/year1_f1_assembly_v2_genbank/HG02630.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02630/assemblies/year1_f1_assembly_v2_genbank/HG02630.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02717/assemblies/year1_f1_assembly_v2_genbank/HG02717.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02717/assemblies/year1_f1_assembly_v2_genbank/HG02717.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02886/assemblies/year1_f1_assembly_v2_genbank/HG02886.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG02886/assemblies/year1_f1_assembly_v2_genbank/HG02886.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03453/assemblies/year1_f1_assembly_v2_genbank/HG03453.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03453/assemblies/year1_f1_assembly_v2_genbank/HG03453.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03516/assemblies/year1_f1_assembly_v2_genbank/HG03516.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03516/assemblies/year1_f1_assembly_v2_genbank/HG03516.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03540/assemblies/year1_f1_assembly_v2_genbank/HG03540.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03540/assemblies/year1_f1_assembly_v2_genbank/HG03540.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03579/assemblies/year1_f1_assembly_v2_genbank/HG03579.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC/HG03579/assemblies/year1_f1_assembly_v2_genbank/HG03579.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG002/assemblies/year1_f1_assembly_v2_genbank/HG002.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG002/assemblies/year1_f1_assembly_v2_genbank/HG002.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG005/assemblies/year1_f1_assembly_v2_genbank/HG005.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG005/assemblies/year1_f1_assembly_v2_genbank/HG005.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG00733/assemblies/year1_f1_assembly_v2_genbank/HG00733.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG00733/assemblies/year1_f1_assembly_v2_genbank/HG00733.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG01109/assemblies/year1_f1_assembly_v2_genbank/HG01109.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG01109/assemblies/year1_f1_assembly_v2_genbank/HG01109.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG01243/assemblies/year1_f1_assembly_v2_genbank/HG01243.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG01243/assemblies/year1_f1_assembly_v2_genbank/HG01243.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02055/assemblies/year1_f1_assembly_v2_genbank/HG02055.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02055/assemblies/year1_f1_assembly_v2_genbank/HG02055.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02080/assemblies/year1_f1_assembly_v2_genbank/HG02080.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02080/assemblies/year1_f1_assembly_v2_genbank/HG02080.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02109/assemblies/year1_f1_assembly_v2_genbank/HG02109.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02109/assemblies/year1_f1_assembly_v2_genbank/HG02109.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02145/assemblies/year1_f1_assembly_v2_genbank/HG02145.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02145/assemblies/year1_f1_assembly_v2_genbank/HG02145.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02723/assemblies/year1_f1_assembly_v2_genbank/HG02723.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02723/assemblies/year1_f1_assembly_v2_genbank/HG02723.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02818/assemblies/year1_f1_assembly_v2_genbank/HG02818.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG02818/assemblies/year1_f1_assembly_v2_genbank/HG02818.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03098/assemblies/year1_f1_assembly_v2_genbank/HG03098.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03098/assemblies/year1_f1_assembly_v2_genbank/HG03098.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03486/assemblies/year1_f1_assembly_v2_genbank/HG03486.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03486/assemblies/year1_f1_assembly_v2_genbank/HG03486.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03492/assemblies/year1_f1_assembly_v2_genbank/HG03492.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/HG03492/assemblies/year1_f1_assembly_v2_genbank/HG03492.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA18906/assemblies/year1_f1_assembly_v2_genbank/NA18906.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA18906/assemblies/year1_f1_assembly_v2_genbank/NA18906.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA19240/assemblies/year1_f1_assembly_v2_genbank/NA19240.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA19240/assemblies/year1_f1_assembly_v2_genbank/NA19240.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA20129/assemblies/year1_f1_assembly_v2_genbank/NA20129.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA20129/assemblies/year1_f1_assembly_v2_genbank/NA20129.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA21309/assemblies/year1_f1_assembly_v2_genbank/NA21309.paternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/NA21309/assemblies/year1_f1_assembly_v2_genbank/NA21309.maternal.f1_assembly_v2_genbank.fa.gz
s3://human-pangenomics/working/HPRC_PLUS/CHM13/assemblies/chm13.draft_v1.1.fasta.gz
s3://human-pangenomics/working/HPRC_PLUS/GRCh38/assemblies/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.gz

The HPRC year one release sequence and pre-built index: <https://giab-data.s3.amazonaws.com/PGR-TK-Files/pgp-tk-HGRP-y1-evaluation-set-v0.tar>

All GFA files, fetched sequences from HPRC year one release of the 385 CMRG: https://giab-data.s3.amazonaws.com/PGR-TK-Files/CMRG_output_dir_v0.3.3.tar
Example Notebooks using PGR-TK including code making most of (the source of) the plots in this manuscript: <https://github.com/Sema4-Research/pgp-tk-notebooks>

Information about a docker image with prebuilt PGR-TK library and Jupyter Lab Server and Usage: <https://github.com/Sema4-Research/pgp-tk/blob/main/pgp-tk-workstation/Readme.md>

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

Life sciences Behavioural & social sciences Ecological, evolutionary & environmental sciences

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Not applicable, our work is on bioinformatics methods not control-case study. The amount data we used are just limited by the amount of genome generated by HPRC or other agencies and computational resource available to us.
Data exclusions	not applicable, see above.
Replication	Not applicable, see above. (The code is totally deterministic, although some code / algorithm improvement may changes the results. Once the code revision is fix, with the same input data, we will get the same output results.)
Randomization	Not applicable, Our results does not dependent objective statistic analysis.
Blinding	Not applicable, Our results does not dependent objective statistic analysis.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involvement in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies
<input checked="" type="checkbox"/>	<input type="checkbox"/> Eukaryotic cell lines
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology
<input checked="" type="checkbox"/>	<input type="checkbox"/> Animals and other organisms
<input checked="" type="checkbox"/>	<input type="checkbox"/> Human research participants
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern

Methods

n/a	Involvement in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input checked="" type="checkbox"/>	<input type="checkbox"/> MRI-based neuroimaging