Annual Review of Genomics and Human Genetics

Pangenome Graphs

Jordan M. Eizenga,[1] Adam M. Novak,[1] Jonas A. Sibbesen,[1] Simon Heumos,[2] Ali Ghaffaari,[3,4,5] Glenn Hickey,[1] Xian Chang,[1] Josiah D. Seaman,[6,7] Robin Rounthwaite,[1] Jana Ebler,[3,4,5] Mikko Rautiainen,[3,4,5] Shilpa Garg,[8,9] Benedict Paten,[1] Tobias Marschall,[3,4] Jouni Sirén,[1] and Erik Garrison[1]

[1]Genomics Institute, University of California, Santa Cruz, California 95064, USA; email: erik.garrison@ucsc.edu

[2]Quantitative Biology Center, University of Tübingen, 72076 Tübingen, Germany

[3]Center for Bioinformatics, Saarland University, 66123 Saarbrücken, Germany

[4]Max Planck Institute for Informatics, 66123 Saarbrücken, Germany

[5]Saarbrücken Graduate School for Computer Science, Saarland University, 66123 Saarbrücken, Germany

[6]Royal Botanic Gardens, Kew, Richmond TW9 3AB, United Kingdom

[7]School of Biological and Chemical Sciences, Queen Mary University of London, London E1 4NS, United Kingdom

[8]Departments of Genetics and Biomedical Informatics, Harvard Medical School, Boston, Massachusetts 02215, USA

[9]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, Massachusetts 02215, USA

Annu. Rev. Genom. Hum. Genet. 2020. 21:139–62

First published as a Review in Advance on May 26, 2020

The *Annual Review of Genomics and Human Genetics* is online at genom.annualreviews.org

https://doi.org/10.1146/annurev-genom-120219-080406

Keywords

pangenome, genome graph, variation graph

Abstract

Low-cost whole-genome assembly has enabled the collection of haplotype-resolved pangenomes for numerous organisms. In turn, this technological change is encouraging the development of methods that can precisely address the sequence and variation described in large collections of related genomes. These approaches often use graphical models of the pangenome to support algorithms for sequence alignment, visualization, functional genomics, and association studies. The additional information provided to these methods by the pangenome allows them to achieve superior performance on a variety of bioinformatic tasks, including read alignment, variant calling, and genotyping. Pangenome graphs stand to become a ubiquitous tool in genomics. Although it is unclear whether they will replace linear

reference genomes, their ability to harmoniously relate multiple sequence and coordinate systems will make them useful irrespective of which pangenomic models become most common in the future.

1. INTRODUCTION

A pangenome models the full set of genomic elements in a given species or clade. Pangenomics thus stands in contrast to reference-based genomic approaches, which relate sequences to a particular consensus model of the genome (**Figure 1**). Genomes that are reconstructed with the aid of a reference genome can appear to be more similar to the reference than they actually are. Pangenomic reference systems can reduce this bias by enabling a new genome to be directly related to all those represented in the pangenome.

Pangenomics has been important to microbiology, where genomic plasticity and diversity have made it indispensable (140), and it has increasingly seen application to eukaryotic genomes (22, 46, 106). Standard pangenomic analyses focus on the presence or absence of genes from given strains and the determination of a core (commonly present) and accessory (frequently absent) pangenome (107). However, they have tended to pay less attention to variation among these sequences and do not typically attempt to provide a precise model relating many genomes to each other at the base-pair level.

By contrast, most high-throughput analyses of large genomes depend on comparison to a single reference genome. However, in recent years, reduced sequencing and de novo assembly costs have supported the discovery of significant levels of large-scale genomic variation in many eukaryotic species, including humans (8, 24, 57, 134), *Arabidopsis thaliana* (2), brewer's yeast (144), and fruit flies (25). These observations have generated wide interest in extending bioinformatic operations to use a pangenomic reference model (32). The availability of true pangenomic references for humans (28) and other model organisms will increasingly render the use of a single reference genome suboptimal. However, using pangenomes effectively requires the development of new bioinformatic methods capable of constructing, querying, and operating on them.

In this review, we consider an emerging class of bioinformatic methods that let us consider genetic diversity at every stage of analysis. We focus primarily on methods that achieve this result by replacing a linear reference genome system with a graphical, pangenomic one. To provide background, we first consider how limitations of the current dominant genome inference paradigm motivate these methods (Section 2). Then, we describe pangenomic models, with a particular focus on graphical ones (Section 3). This provides a foundation to understand how they can be constructed from sequencing data or assembled genomes (Section 4). We then survey index data structures that allow us to interact with them efficiently (Section 5). These index structures allow us to interrogate, visualize, and relate new information to pangenomes, such as in the process of read alignment (Section 6), yielding results in a variety of new data formats (Section 7). Finally, we examine several downstream applications of these models (Section 8) and reflect on the future impact of these methods on genomics (Section 9).

2. RESEQUENCING RECONSIDERED

Biological research frequently depends on our ability to infer the sequence-level relationships between genomes. In an earlier era, when many sequencing studies focused on small regions or single genes, it was often feasible to relate all relevant sequences to each other. Although limited

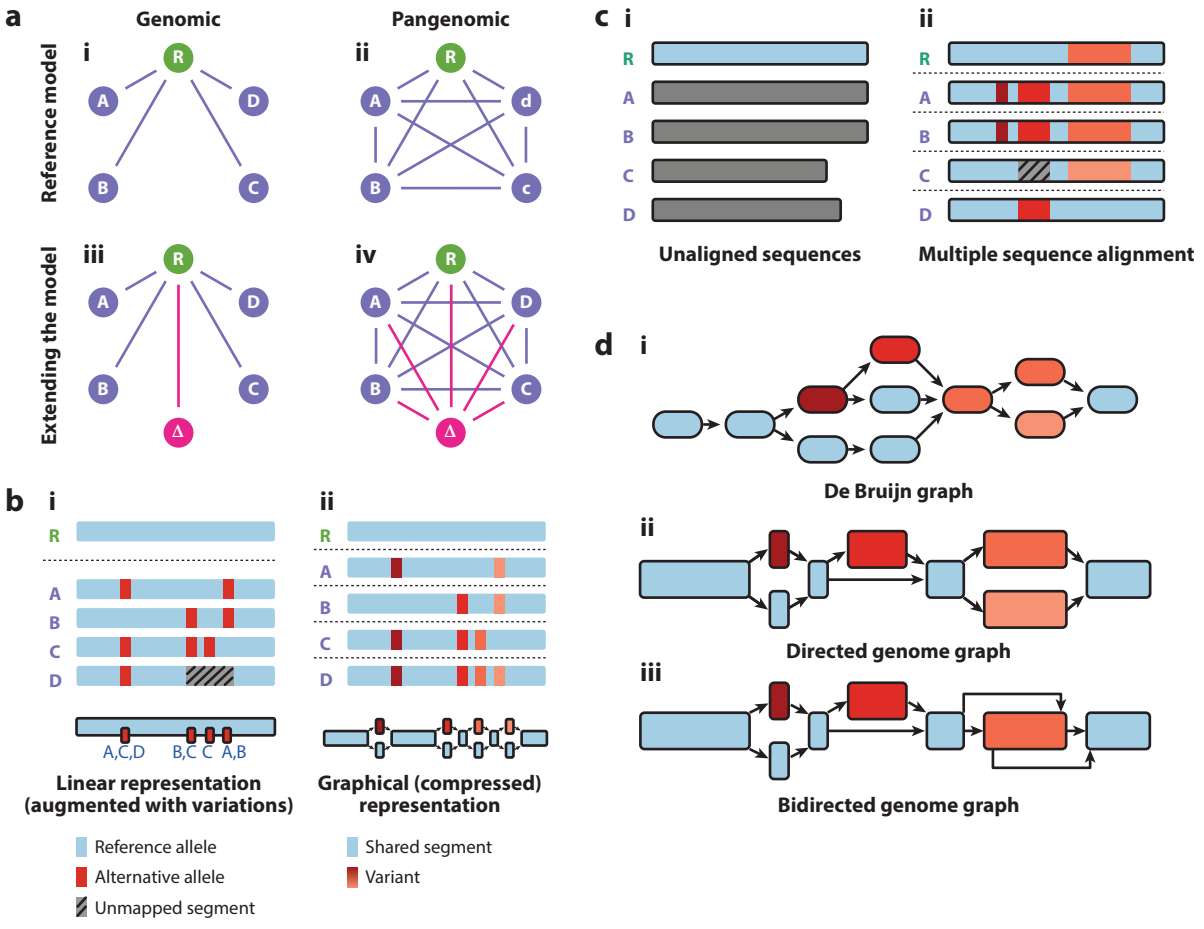

Figure 1

Pangenomic models. (*a*, *i*) In reference-based genomic analyses, all genomes (A–D) are compared with each other via their relationship to the reference genome (R). (*ii*) In a pangenomic setting, one attempts to model direct relationships between all the genomes in the analysis, from which a particular reference is chosen arbitrarily. (*iii*) When extending the analysis with a new genome, Δ, one adds it to the genomic model by comparing it with the reference genome. (*iv*) By contrast, adding a new genome to a pangenomic analysis compares it directly with all other genomes in the model. (*b*, *i*) Regions of some genomes are unalignable against the reference and cannot be represented in a list of variants. (*ii*) A graphical model of the genomes allows a direct all-to-all comparison, capturing all of their sequence relationships. (*c*, *i*) A collection of sequences representing a pangenome. (*ii*) Multiple sequence alignment of the sequences captures their mutual relationships. (*d*, *i*) In a de Bruijn graph, sequences are represented without bias, but variants may correspond to larger graph structures. (*ii*) An acyclic sequence graph is equivalent to the multiple sequence alignment. (*iii*) A generic sequence graph can compactly represent a structural variant (shown in *orange*), using edges between the forward and reverse strands of the graph to indicate the presence of an inversion.

in scope, these analyses were effectively pangenomic. They were based on many-to-many relationships between sequences, typically derived by multiple sequence alignment. Precise multiple sequence alignment methods are expensive, with popular algorithms scaling cubically with the number of input sequences (102). It is infeasible to apply such computationally demanding methods to the data scales obtained with modern high-throughput sequencing. Instead, high-quality genome assemblies and high-throughput sequencing have encouraged resequencing methodologies, wherein reads from each sample are aligned to a single reference genome. This approach

is practical and scalable. State-of-the-art resequencing pipelines can jointly analyze tens of thousands of genomes (111) at a cost per genome that is only a small fraction of the total sequencing cost.

Although efficient and conceptually simple, resequencing has a significant limitation. Precise genomic relationships are visible only for sequences that are similar enough to the reference genome to be alignable (**Figure 1**), an effect known as reference bias. It is strongest for structural variation or sequences that are absent from the reference system (134), but it can be relevant even for single-nucleotide polymorphisms (SNPs), which causes problems in allele-specific expression quantification (23) and in the analysis of ancient DNA (146). Given that this bias shapes the methods used to establish models of the truth (147), it is difficult to even evaluate without paradigmatic change in our analysis techniques.

Estimates based on short-read sequencing data have placed the human pangenome at between 1% (84) and 10% (126) larger than the GRCh38 human reference assembly. Others have demonstrated that up to several megabase pairs of sequence are present in each new individual and not in the reference (8, 57). We expect that whole-genome telomere-to-telomere assemblies based on long single-molecule sequencing will provide greater insight into the extent, placement, and significance of these novel sequences (76, 92). Even if we know of their existence, the reference bias inherent in resequencing will continue to limit the ability of researchers to relate new sequences to these regions. Resequencing methods that use a pangenomic reference system should not be subject to this limitation.

3. PANGENOMIC MODELS

A pangenomic model (**Figure 1**) is a data structure that represents the genomic sequences of a population, a species, a clade, or even a metagenome (32). The model serves as a central coordinating entity to describe the collection of sequences and genomes in the pangenome. Pangenomic models may take many forms, including collections of unaligned sequences or learned sequence models, but here we focus mostly on graphical ones.

Sequence graphs serve to compress many redundant input sequences into a smaller data structure that is still representative of the full set (58). Sequence graphs may have their nodes or edges labeled with DNA sequences, but for simplicity we focus on the node-labeled case. In a node-labeled sequence graph, edges indicate when concatenations of the nodes they connect occur in the sequences modeled by the graph. Walks through a sequence graph thus include the set of sequences from which it was built. These graphs are referred to as bidirected when they represent both strands of DNA and inversions between them. Sequence graphs were first used to represent multiple sequence alignments (58, 77). In assembly, they have been applied to represent the full information in a set of sequencing reads (as in a string graph) (97) or fixed-length *k*-mers (as in a de Bruijn graph) (110).

Genome graphs are sequence graphs used to represent whole-genome relationships (109). Walks through these graphs represent recombinations of the genomes included in the model. Regions of the graph where multiple paths connect a common head and tail node, often referred to as bubbles (108), represent variation. Variation graphs further structure this model by embedding the linear sequences of the pangenome as paths (50). [Variation graphs are similar to the variant graphs used in textual research to model a collection of revisions of the same text (123).] Paths provide a stable coordinate system that is unaffected by the manner in which the graph was built, thus supporting the coordination of positions, annotations, and alignments between variation graphs and linear reference genomes.

4. BUILDING A PANGENOME

Methods to construct pangenomic data structures mirror the classes of pangenomic models. A pangenome may simply be a collection of sequences, in which case construction is similar to the genome or metagenome assembly problem, or it may include information about the alignment of sequences or genomes within it. This alignment could be compressed into a set of variants found against a set of reference sequences. If this alignment is based on *k*-mers, then it implies a de Bruijn graph. If it is a complete, gapped alignment, covering small and large variation, then the pangenome model can be thought of as a whole-genome alignment.

4.1. Collecting Sequences

A pangenome can be represented as a collection of sequences. Several approaches support the construction, annotation, and interrogation of these pangenomic sequence collections. Panseq (75) finds novel regions, determines the core and accessory genome, finds SNPs within the core pangenome, and then determines a subset of loci useful for molecular fingerprinting. PGAP (Pan-Genomes Analysis Pipeline) (145) extends Panseq's approach with modules for evolutionary and functional analysis and is implemented as a single stand-alone executable. Recent work has focused on scaling these techniques to ever larger genomes. PanTools (125) detects and annotates homology groups in large collections of large genomes using a *k*-mer-based approach. Its detailed graph database model connects the panproteome defined by homology groups to genomic annotations and sequences. HUPAN (Human Pan-Genome Analysis) (38) extends the sequence collection model to human and large eukaryotic genomes, taking assembled genomes as input and finding nonreference sequences within them by comparison to a reference genome.

4.2. Adding Variation

Rather than collecting unique sequences that represent a collection of genomes, we can consider small variants between the collection and a reference genome. Such a model directly implies a directed acyclic graph, ordered along the reference genome, with bubbles at the sites of variation. This pangenome construction approach is used in diverse graph genome read mappers, including GenomeMapper (124), Seven Bridges' Graph Genome Pipeline (115), PanVC (Pan-Genomic Variant Calling) (139), and Gramtools (87). The VG (Variation Graph) toolkit, specifically VG construct (50), can be applied to transform VCF (Variant Call Format) files and reference sequences into genome graphs. Some methods, such as the journaled string tree (114), and methods based on elastic degenerate texts (12), such as SOPanG (Shift-Or for Pan-Genome) (29), transform the variant set and reference into a structure optimized for online sequence queries of the pangenome. Deciding which variation should be added to a graph is nontrivial and has encouraged studies of graph utility (104) and algorithms to determine which variation is helpful (112).

4.3. Colored, Linked, and Compacted de Bruijn Graphs

De Bruijn graph–based assemblers can be given a pangenomic quality through the addition of colors to their nodes (*k*-mers) or unitigs (unbranching components in the graph). Each color provides a mapping between a specific biosample and a subset of the graph. Cortex first demonstrated that colored de Bruijn graphs could perform population-scale analyses with an efficient graph implementation (65). Recent improvements to colored de Bruijn graph construction, such as Bifrost (62), allow the construction of colored de Bruijn graphs from very large sequence sets (the authors built a pangenome of 118,000 *Salmonella* genomes) and further support efficient updates of

these pangenomic models. The feature that makes these methods efficient—the fixed k on which they are based—also limits their resolution of repetitive genomic features. It is not feasible to build them from noisy third-generation sequencing reads. Addressing these limitations, several methods embed linking information within the de Bruijn graph that can be used to reconstruct embedded haplotypes or reads (15, 137).

Several methods use compacted de Bruijn graph construction to elaborate pangenome graphs (here, compacted means that chains of k-mers that contain no internal furcations are merged into a single node in the graph representation). SplitMEM (Split Maximal Exact Matches) (90) uses a suffix tree with suffix skips to derive the set of maximal exact matches $\geq k$ between a set of genomes. Improving on this result in both time and space efficiency, Baier et al. (11) demonstrated two similar pangenome graph induction algorithms based on succinct representations of the suffix tree and the Burrows–Wheeler transform (BWT) (21). TwoPaCo (95) applies a probabilistic data structure to narrow the set of candidate vertexes in the compacted de Bruijn graph of a set of genomes, supporting the efficient generation of a pangenome graph from larger genomes than previous methods.

4.4. Alignment-Based Sequence Graphs

Sequence graphs (58) can be understood as representations of the mutual alignment of a set of sequences. Alignment-based pangenome structures form the basis of several pangenomic approaches. They have found use in the construction of acyclic multiple sequence alignments. POA (Partial Order Aligner) (53, 77) uses an acyclic, directed sequence graph model to build multiple sequence alignments. ProgressiveCactus (7) produces whole-genome alignments that can be rendered as sequence graphs. SibeliaZ (94) finds collinear blocks within TwoPaCo's compacted de Bruijn graph and applies POA to each to yield a whole-genome alignment graph. VG msga (Variation Graph multiple sequence/graph aligner) (48, 50, 104) generalizes the progressive approach of POA to build generic variation graphs that include cycles and inversions.

Not all researchers have focused on generic graphs, with several arguing that completely generic models are either computationally intractable or not relevant to important practical analyses. REVEAL (Recursive Exact-Matching Aligner) (85) builds a pangenome graph from a syntenic set of maximal exact unique matches of decreasing size between a pair of sequences (or graphs) and later adds inversions detected by alignment of paths through bubbles in this graph. NovoGraph (14) follows a reference-guided approach, breaking a set of genomes into syntenic alignable blocks, deriving a multiple sequence alignment for each, and yielding a VCF file as its output. Similarly, seq-seq-pan (68) employs existing whole-genome alignment methods to find a set of locally collinear blocks, which it compacts into a sequence graph that respects the synteny of the input genomes. GenGraph (4) realigns previously identified collinear blocks, yielding a genome graph from a multiple sequence alignment.

Recent unpublished methods explore two new alternatives to alignment-based pangenome construction. Minigraph (**https://github.com/lh3/minigraph**) extends the minimap2 (83) alignment chaining model to work on graphs. It applies this alignment model to progressively build out a pangenome graph from a series of genomes that contains large sequences (>250 base pairs) that were not previously seen in other genomes. The resulting pangenome does not contain all input sequences and variation between them but rather a representative subset and large structural variants. By contrast, seqwish (48; **https://github.com/ekg/seqwish**) generates the full variation graph implied by a collection of sequences and alignments between them. The paths embedded in its output graph precisely and completely reconstruct the input sequences, while the topology of the graph describes all variants represented in the input alignments.

4.5. Positional Systems in Pangenomes

Reference genome sequences provide a coordinate system to catalog and exchange information about genes, protein binding sites, epigenetic profiles, variants, and homologies. In linear references, genomic coordinates are easily interpretable, and they unambiguously indicate both the layout of the sequence and the distance between bases, but this is not the case when these coordinates are embedded within a graph (116).

It is possible to use reference coordinates in a graphical pangenome by embedding reference sequences inside the graph and labeling graph nodes with their relative positions in these paths (48, 50). This approach has been extensively explored in variation graph–based tools. However, several problems remain. The embedded coordinate systems may be incomplete, in that they may not fully cover the graph. Also, particular graph instantiations may induce ambiguity in reference positions. For instance, a copy number variant that is collapsed in the graph will contain multiple overlapping reference path coordinate ranges.

These limitations have driven the development of complete coordinate systems for genome graphs. One solution is to build a hierarchy of graph components, based on a starting reference sequence, adding a new name and coordinate range for each nonreference sequence that is included (minigraph uses a similar model) (116). Another technique is to build positional systems based solely on the topology in the graph (108). Genomic variation creates a system of nested bubble structures that can be used to spatially organize graph elements. This approach has a rigorous, if complex, mathematical basis. Similar decompositions of the graph topology have been used in assembly-based variant detection (65, 105).

5. INDEXING PANGENOMES

Index data structures for pangenome graphs support efficient random access to elements and features of the graph. Attention must be given to ensure that these index structures do not require significant overhead relative to the information content of the graph. Naive implementations of sequence and structural indexes of the graph can incur significant run time and memory costs, which can become problematic as graph sizes increase. Succinct data structures and careful encoding of these data are thus required to reliably fit large graphs into the main memory of commodity computing systems. Particular index models lie at the core of the highest-performing graph-based visualization (Section 6.1), read mapping (Section 6.3), and variant calling systems (Section 8.1).

Building text indexes is more involved for sequences encoded in a graph than for linear references. In graphs with regions of dense variation, the number of k-base-pair paths can grow exponentially with k, often rendering their complete enumeration intractable even for low values of k. To limit the exponential growth, the index may support only relatively short query strings. Some indexes (131) support longer queries by doing extensive preprocessing. In others (63, 87, 136), queries mapping to complex graph regions can be slow. Instead of indexing the entire graph, the index may contain only k-mers from a simplified graph or from specific paths of the graph.

5.1. Indexing Sequences Using a Graph

The FM index (full-text index in minute space) (42) is a text index, based on the BWT (21), that is frequently used with DNA sequences. One variant of the FM index, RLCSA (Run-Length Compressed Suffix Array) (88), run-length encodes the BWT, allowing it to store and index a collection of similar sequences in a space-efficient way. If we know a good global alignment of the sequences, we can use that information to make the index both smaller and faster (64). This approach was developed further in the FM index of alignment (99, 100). Both Huang et al. (64)

and Na et al. (100) used the graph induced by the alignment as a space-efficient representation of the sequences.

5.2. Indexing Acyclic Graphs

One class of graph indexing methods supports only acyclic graphs, often represented as directed acyclic graphs. This constraint can exist either because the acyclicity of the graph provides guarantees that simplify the problem or because incidental features of the method's software implementation preclude use on cyclic graphs.

GenomeMapper (124), the first graph-based read aligner, was limited to such graphs. Its indexing was also relatively simple. GenomeMapper uses a simple hash-based k-mer index, with $k \leq 13$ to limit memory usage.

GCSA (Generalized Compressed Suffix Array) (131) was the first attempt to generalize the BWT for graphs. It applies several graph transformations that preserve the graph's sequence space while creating an unambiguous ordering for nodes. When the graph's complexity is low, these transformations are reasonably fast and do not increase the size of the graph significantly. However, at a certain threshold of variant density, the transformed graph quickly becomes too large to handle.

BWBBLE (63) is a BWT-based representation for VCF-based pangenome graphs. Simple substitutions are encoded in the sequence using International Union of Pure and Applied Chemistry (IUPAC) codes, and the sequence is indexed using a normal FM index. Because each base can be encoded using eight different characters, the search branches at every base to cover all possible characters that admit the base searched. In practice, most branches quickly run out of results and can be pruned from the search. BWBBLE represents insertions and deletions with extra sequences, including a given amount of context around the variant. The length of this context is an effective upper bound for query length.

The vBWT (variation Burrows–Wheeler transform) (87) took another approach to using the BWT for indexing VCF-based pangenome graphs. It encodes variants as `(ref|alt1|alt2|...)` in the sequence. When the search encounters a variant, it must branch to handle each allele separately. Both BWBBLE and the vBWT trade faster index construction for slower queries. However, a combination of IUPAC codes for substitutions, the vBWT approach for other variants, and a k-mer index for matching the first 5–10 bases is faster than either of the originals (20).

5.3. General Graphs

Some text indexes are based on Lempel–Ziv parsing or context-free grammars. These indexes first find partial matches between the query string and the indexed phrases and then combine the partial matches into full matches using two-dimensional range queries. In the hypertext index (136), each node is a separate phrase. Queries mapping to a single node or crossing a single edge can be matched efficiently, while finding mappings to complex graph regions can be slow.

Techniques similar to GCSA can be used to represent de Bruijn graphs (16). If the graph transformations used in GCSA construction are stopped after i steps, the resulting graph is equivalent to an order-2^i de Bruijn graph. This de Bruijn graph can be used to approximate the original graph. Queries of this index yield no false negatives, but matches longer than 2^i may be false positives. By using this approach, GCSA2 (129) attempts to support fast queries in arbitrary graphs.

GCSA2 faces the same issues with complex graphs as GCSA. In practice, most graphs must be simplified before they can be indexed. Typical simplifications include removing high-degree nodes and complex regions from the graph and replacing them with the reference sequence. If a collection of haplotypes is available, the removed regions can be replaced with new subgraphs

that contain separate paths for each distinct local haplotype (130). This way, the index contains all *k*-mers from the haplotypes, while usually missing some *k*-mers from their recombinations.

5.4. Indexing Graphs Using Sequences

Instead of attempting to index the entire graph, it is often sufficient to index only selected paths in it. CHOP (96) takes the paths corresponding to haplotypes and breaks them into smaller pieces. The distinct pieces form an artificial linear reference, which can be used with any read aligner. The process guarantees that any substring of the haplotypes of length *k* is also a substring of one of the pieces. As with BWBBLE, *k* represents an effective upper bound for query length.

PSI (Pan-Genome Seed Index) (51) follows a similar approach with artificial paths. Instead of using haplotypes, PSI uses a greedy algorithm to find a set of paths that covers as many *k*-base-pair windows in the graph as possible. When a fully sensitive index is needed, PSI can reverse the role of the query strings and the graph. While complex graph regions may contain an excessive number of *k*-mers, the reads mapping to them contain only a limited number of *k*-mers. By indexing a batch of reads and searching for the complex regions in that index, all mappings of the query strings to the graph can be found with reasonable resources.

5.5. Indexing Haplotypes and Genomes in Variation Graphs

Haplotypes in related individuals are typically highly similar and thus compressible. A series of results lead from a compact haplotype index for biallelic SNPs to a generic haplotype index for complex variation graphs. The PBWT (positional Burrows–Wheeler transform) (39) provides an efficient compressed representation of a set of haplotypes over biallelic variable sites. Like the BWT, it supports efficient haplotype matching queries, such as maximal exact match finding. Later work (45) showed that the PBWT is equivalent to the wavelet matrix (30), which is the most efficient known encoding of strings with large alphabets supporting a variety of important random access queries. The gPBWT (graph positional Burrows–Wheeler transform) (103) extends the PBWT model to haplotype walks embedded in complex sequence graphs. The GBWT (graph Burrows–Wheeler transform) (130) builds on several assumptions that tend to hold for sequence variation graphs to improve run time and memory costs relative to the gPBWT. Provided that the variation graph on which they are built encodes the full complement of variation in the pangenome, both the gPBWT and GBWT are excellent, and in principle lossless, compressors of genomes. A GBWT for the 5,008 haplotypes in the 1000 Genomes Project required 14.6 GB, or approximately 1 bit per 1 kilobase pair of encoded genomic sequence.

6. RELATING NEW INFORMATION TO THE PANGENOME

Here we focus on two major avenues to interact with pangenome graphs. By visualizing these graphs, we can gain insight into the relationship between genomes, learning about variation small and large. Pangenome models are often used as a reference system for sequence alignment, which enables the relation of a new biosample to the pangenome to support a wide variety of downstream applications.

6.1. Visualization

The visualization of pangenome graphs presents significant challenges that have proven to be difficult to resolve in a single system. **Table 1** presents an overview of such tools, while **Figure 2** compares different visualization methods applied to the same graph.

Table 1 Overview of graph visualization tools

Tool[a]	Layout[b]	Graph type[c]	Proven scale	Extra views[d]	UI[e]	Back end[f]
Bandage (141)	FD	A	100 Mbp	None	App	OGDF (27)
GfaViz (52)	FD	A, V	1 Mbp	None	App	OGDF (27)
SGTK (73)	FD	A, S	1 Mbp	B, L	Web	cytoscape.js (43)
AGB (93)	Rank	A	10,000 edges[g]	L	Web	d3-graphviz (**https://github.com/magjac/d3-graphviz**)
Sequence Tube Map (13)	Tube	V	100 Mbp	B, L	Web	NA
MoMI-G (143)	Tube, Circos	V	1 Gbp	B, L	Web	Sequence Tube Map (13), Circos (72)
VG view (50)	Rank	V	10 Kbp	B	CLI	GraphViz
VG viz (48)	SM	V	100 Kbp	B, L	CLI	NA
ODGI viz	SM	V	1 Gbp	B, L	CLI	NA

[a]AGB, Assembly Graph Browser; GfaViz, Graphical Fragment Assembly Visualization; MoMI-G, Modular Multiscale Integrated Genome Graph Browser; ODGI viz, Optimized Dynamic Genome Graph Implementation visualization; SGTK, Scaffold Graph Toolkit; VG view, Variation Graph view; VG viz, Variation Graph visualization.
[b]FD, force-directed layout; rank, GraphViz-style rank-based layout; SM, sorted-matrix layout.
[c]A, assembly graph; S, scaffold graph; V, variation graph.
[d]B, base-level view; L, linear view.
[e]UI, user interface; app, native application; web, browser-based interface; CLI, command-line tool.
[f]OGDF, Open Graph Drawing Framework; NA, not applicable.
[g]See **https://github.com/almiheenko/almiheenko.github.io/blob/8f4b2f8c/AGB/Flye_Human/data/repeat_graph.json**.

Complex, nonlinear graph structures are difficult to present in a convenient number of dimensions. Traditional genome browsers organize information in a two-dimensional rectangle, with a linear reference's sequence space providing the horizontal dimension and a stack of annotations plotted in the vertical dimension (56). When visualizing a graph, there are two basic approaches: One can either attempt to mimic linear browsers and compress the graph into one dimension, or give the graph two dimensions and represent annotations using interactivity, labels, or color. Both options are challenging, as pangenome graphs are often nonplanar, requiring at least three dimensions to draw without overlapping lines. Force-directed layout, an optimization-based graph layout method, is a popular approach in the second category, but displaying annotations in this layout is challenging.

Visualization tools can be categorized by the kinds of graphs they are intended to display, in addition to whether they can linearize a graph. Many tools, including Bandage (141) and GfaViz (Graphical Fragment Assembly Visualization) (52), are designed for interpreting assembly graphs. These tools tend to focus on displaying the overall, rather than base-level, structure of the graph and have no visual features to reflect the pangenomic aspect of the graph.

Tools initially designed for variation graph visualization, on the other hand, tend to focus more on base-level structure and pangenomic relationships. For example, Sequence Tube Map (13) displays precise base-scale variation, haplotypes, and short-read mapping locations using a visual language inspired by transit system maps.

There is a difference in scale when moving from an assembly or scaffold graph to a comprehensive variation graph of a species pangenome, and this scale difference also separates different visualization tools. High-level assembly-graph tools such as AGB (Assembly Graph Browser) (93) struggle to show fine details in the graph, while low-level variation graph tools such as VG view (50) and Sequence Tube Map cannot scale to large graphs. One tool that works well at both

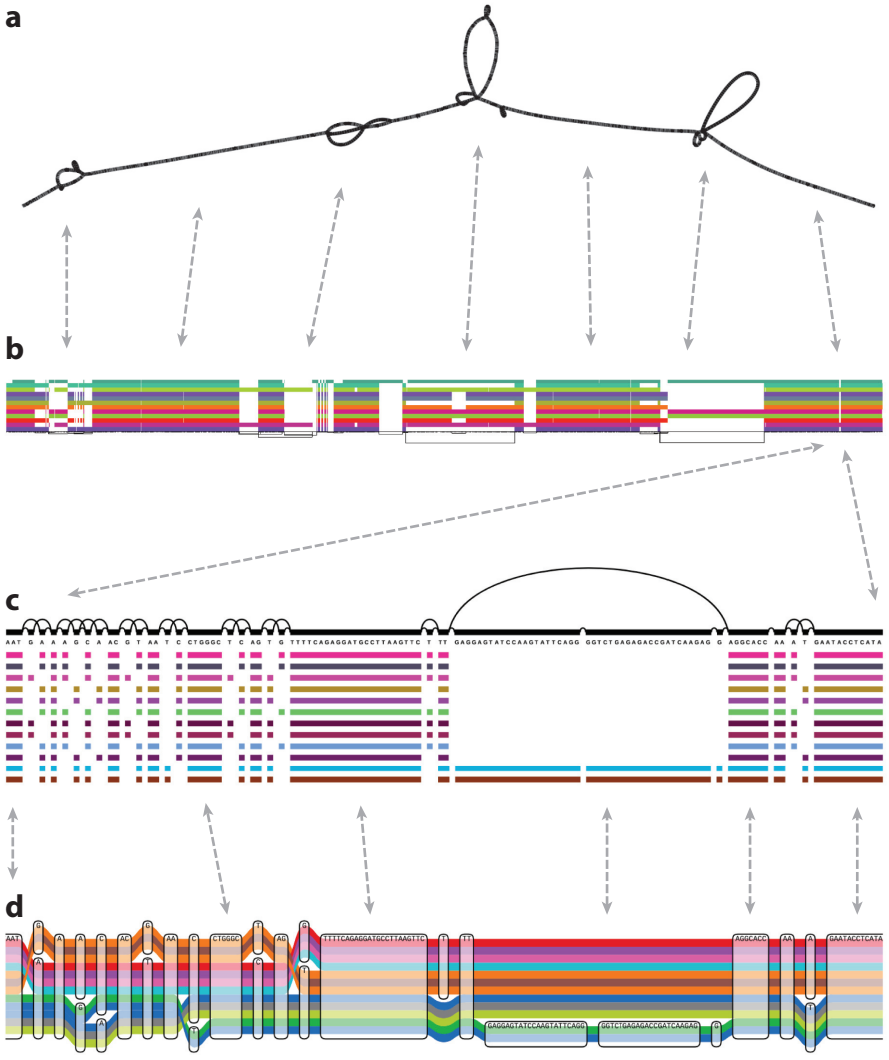

Figure 2

Visualizing a graph of GRCh38 and its alternate sequences in the gene *HLA-DRB1* built with VG msga (Variation Graph multiple sequence/graph aligner) (48). (*a*) Bandage's force-directed layout, revealing large-scale structures (141). (*b*) An ODGI viz (Optimized Dynamic Genome Graph Implementation visualization) binned, linearized rendering of the paths (*colored bars*) versus the sequence and topology of the graph (*thin lines below the bars*). (*c*) A fragment of a VG viz (Variation Graph visualization) linearized rendering, showing base-level detail. (*d*) The same fragment rendered with Sequence Tube Map (13). Dashed lines show the correspondences between the visualizations. Path colors are assigned independently by each method.

large scale and high detail is MoMI-G (Modular Multiscale Integrated Genome Graph Browser) (143). Designed as a multiscale graph browser, it presents both a Sequence Tube Map rendering of base-level differences and a Circos plot (72) of chromosomal-scale connections and can uniquely visualize long reads in the context of pangenomic haplotypes (143). ODGI viz (Optimized Dynamic Genome Graph Implementation visualization) (**https://github.com/vgteam/odgi**) uses

binning and direct rendering to a raster image to generate visualizations representing gigabase-scale pangenomes. The approach is a rasterized version of the linear layout technique of VG viz (Variation Graph visualization) (48).

Interactively visualizing human-genome-scale, human-pangenome-detail graphs coherently across zoom levels remains an open problem.

6.2. Graph Alignment Algorithms

Sequence comparison is at the core of many genomic analyses, and sequence alignment is the essential method for doing so. Classic algorithms such as the Smith–Waterman algorithm (132) do not directly apply to genome graphs. However, the recurrence relations that drive their scoring and traceback routines can be extended to allow the alignment of sequences to acyclic sequence graphs, as popularized by POA (77). Further generalizations support the alignment of sequence graphs to sequence graphs (53), sequences to cyclic graphs (101), and even cyclic sequence graphs to cyclic sequence graphs (5, 98). It is notable that many of these findings have been independently rediscovered or refined by contemporary researchers (6, 67, 74, 118). Some earlier algorithms require restricted scoring functions to achieve efficiency (118), but recent contributions have used less restricted functions that produce more biologically meaningful alignments in some contexts (67).

Graph alignment algorithms have also become faster. POA had an equivalent asymptotic run time to linear alignment but required acyclic graphs (77). Later optimizations simply ran slower on general graphs (69). Algorithms are now known with equivalent run time even on general graphs (67). In addition, researchers have developed modified algorithms that run quickly in the practical context of real-world computer architectures (66, 117, 135).

6.3. Genome Graph Mapping

Efficient methods to map reads to large pangenome graphs have been developed in recent years. Many draw on recent research in alignment (Section 6.2) and advances in pangenome indexing (Section 5).

Although these mapping tools all target sequence graphs, there are significant differences in the types of graphs that they handle (**Table 2**). Several tools apply only to acyclic variation graphs formed by adding variants to a linear reference. Examples include GenomeMapper (124), Seven Bridges' Graph Genome Aligner (115), HISAT2 (Hierarchical Indexing for Spliced Alignment

Table 2 Overview of graph mapping tools

Tool[a]	Graph types	Sequencing types[b]	Other notes
deBGA	De Bruijn graph	NGS	
BGREAT			Gapless alignment
BrownieAligner			
GenomeMapper	Acyclic variation graph	NGS	No longer maintained
Graph Genome Aligner			
HISAT2			Fast
V-MAP		NGS and long read	
VG	Variation graph	NGS and long read	Accurate, high memory usage
GraphAligner	Variation graph or overlap graph	Long read	Fast, high memory usage

[a]BGREAT, de Bruijn Graph Read Mapping Tool; deBGA, de Bruijn Graph–Based Aligner; HISAT2, Hierarchical Indexing for Spliced Alignment of Transcripts 2; VG, Variation Graph; V-MAP, Variant Map.
[b]NGS, next-generation sequencing.

of Transcripts 2) (71), and V-MAP (Variant Map) (138). By contrast, VG (48) and GraphAligner (119) appear to be the only tools with open ambitions of mapping to arbitrary variation graphs, including complex local and global topologies. GraphAligner can also align to generic overlap graphs and de Bruijn graphs, a feature that it uses to drive the error correction of long reads using de Bruijn graphs (44, 60).

The majority of these tools emphasize mapping short-read next-generation sequencing (NGS) data. To our knowledge, GraphAligner and V-MAP are the only graph mapping tools designed for long-read sequencing data (119, 138). While V-MAP also supports NGS reads, GraphAligner's seeding strategy limits it to long reads. VG also supports long-read alignment (50), but this is based on a hierarchical approach that applies the alignment algorithm for short reads to chunks of long reads (48). The approach is accurate but nearly an order of magnitude slower than GraphAligner (119).

For indexing (Section 5), most graph mapping tools have opted for some variation of a *k*-mer table. GraphAligner, GenomeMapper, the Seven Bridges mapper, and V-MAP all use this strategy (115, 119, 124, 138). The remaining mappers use succinct text indexes. VG uses GCSA2 (129) and a longest-common-prefix array, which enable highly specific queries at the expense of high memory utilization (48). HISAT2 uses a modified GCSA (131) that also encodes the graph structure itself, which helps give HISAT2 an impressively low memory footprint but a somewhat more limited set of queries (71). GraphAligner also has the option of seeding with a full text index of the node sequences of the graph, but this option is not enabled by default (119).

Most graph mappers employ graph-based alignment algorithms. The exceptions are GenomeMapper, which aligns to all paths out from a seed, and HISAT2. The HISAT2 alignment algorithm relies on a complex set of heuristics that depend heavily on its exact match index, which makes it exceptionally fast but can also hurt alignment quality around insertions or deletions (indels). VG and V-MAP both employ some version of partial order alignment (48, 138). The Seven Bridges mapper first searches for a near-exact match using an exponential depth-first search and applies partial order alignment if this search fails (115).

Due to the recent development of these methods, there have been few independent comparative studies of their performance and accuracy. In general, VG compares favorably to other tools in terms of accuracy on NGS data (116). However, it requires more memory, has slower indexing, and often maps more slowly than the alternatives (71, 138). V-MAP's fast clustering heuristics allow it to align long reads faster than VG, but it has not been compared with GraphAligner (138). GraphAligner is the only mapper to incorporate the most recent research into graph alignment algorithms. It uses a banded alignment algorithm to achieve impressive speed in aligning long reads to genome graphs (119).

RNA splicing can be represented directly in a genome graph model (77). It follows that graph mappers can be applied to RNA sequencing (RNA-seq) data. HISAT2 can map RNA-seq data in addition to genomic DNA (71). It is based on the RNA mapper HISAT (70) and retains the capacity for spliced alignment. The ability to create spliced variation graphs has also been added to VG (50). In these variation graphs, known splice junctions are added as edges, similar to the addition of a deletion event. VG supports any type of variation, but its splicing awareness is limited to splice junctions represented in the graph. Thus, reads that span a novel splice junction will only map partially. We further discuss applications of graph mapping to functional genomics in Section 8.3.

7. PANGENOMIC DATA FORMATS

Several common data formats are used to exchange pangenomic models. Pangenomes can be stored as collections of sequences in the FASTA format. Variant calls in VCF format (33) may

be added to such a collection to describe small or structural variants found in the pangenome. However, to exchange graphical pangenomes, the community frequently uses a subset of the Graphical Fragment Assembly version 1 (GFAv1) format (**https://github.com/GFA-spec/GFA-spec/blob/master/GFA1.md**). Only a small subset of GFA is required to represent pangenome graphs, but using this format allows pangenomic analyses to use many genome assembly tools.

To represent read alignments to pangenome graphs, the VG tool kit has developed the GAM (Graph Alignment/Map) format (50), which generalizes the SAM (Sequence Alignment/Map)/BAM (Binary Alignment/Map) (59) data model to pangenome graphs. GAM is produced by several other alignment tools (67, 119) and consumed by numerous downstream applications. GAF (Graph Alignment Format) (**https://github.com/lh3/gfatools/blob/master/doc/rGFA.md#the-graph-alignment-format-gaf**) generalizes the text-based PAF (Pairwise Alignment Format) (**https://github.com/lh3/miniasm/blob/master/PAF.md**) to work on graphs encoded in GFA. GAF can also describe mappings to graphs encoded in rGFA (Reference Graph Alignment Format) (**https://github.com/lh3/gfatools/blob/master/doc/rGFA.md**), which is a specialization of GFA for reference pangenome graphs.

8. APPLICATIONS OF PANGENOMIC MODELS

Although graphical pangenomic techniques can be applied throughout biology, most recent work has focused on a handful of applications where they can provide substantial benefits. Reduced reference bias and coherent representations of alleles yield significant improvements in structural variation detection and can decrease the run time costs of genotyping. They have enabled haplotype reconstruction of quasi-species in metagenomic studies and transcripts in functional genomics and supported pangenomic association studies.

8.1. Variant Calling and Genotyping

Typically, variant calling and genotyping indicate different aspects of a reference-guided genome inference process. Genotyping consists of determining whether a previously observed variant is present in a new sample, whereas variant calling involves detecting previously unobserved variation. When the reference system is a linear genome, these two steps are often merged. A single process will detect candidate variation and infer a sample genotype at each putatively variable locus (49, 81).

These methods are often Bayesian and combine a model for observation and error with a simple a priori model of the expected pattern of variation, typically based on the expected rate of heterozygosity or mutation. By using multisample variant calling, genotype phasing, and genotype imputation (1, 18), one can share information between samples to improve the accuracy of a reconstruction of genomes from short reads. However, this joint calling approach is expensive and is not applicable when one has only a few new genomes to reconstruct. Furthermore, it does not help the primary interpretation of new sequencing data during read mapping.

Alignment to any allele in a pangenome reference graph is as efficient as alignment to the reference allele in a linear reference sequence. Including known genetic variation in the pangenomic reference system can thus reduce bias toward the reference allele when genotyping heterozygous variants (**Figure 3**). This effect is strongest for alleles that are highly divergent from the reference, suggesting that the impact of these approaches on variant calling and genotyping will be strongest for large indels and structural variants.

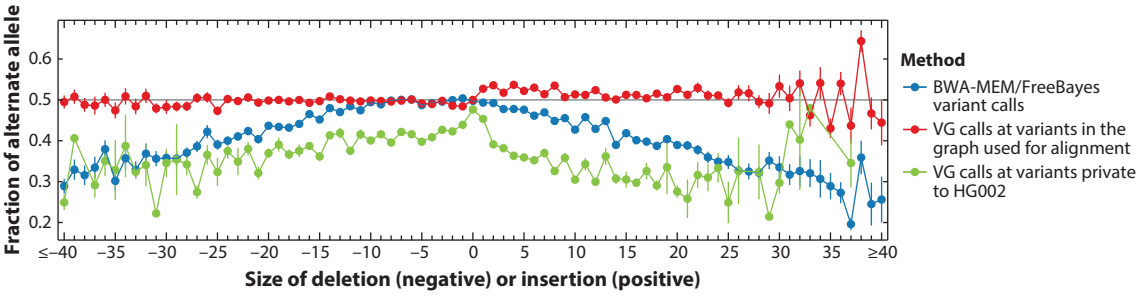

Figure 3

Mean alternate allele fraction at heterozygous variants in the HG002/NA24385 genome sequence validated in the Genome in a Bottle truth set (147) as a function of deletion or insertion size (single-nucleotide polymorphisms at 0). Error bars are ±1 standard error of the mean. Blue points show the allele balance metric across allele lengths for alignments with BWA-MEM (Burrows–Wheeler Aligner Maximal Exact Matches) (82) and variant calls made by FreeBayes (49). Variant calls were made with alignment to a variation graph built from the 1000 Genomes Project variants (1), followed by variant calling in VG (Variation Graph). These calls are divided into two groups: calls at variants in the graph used for the alignment (*red*) and calls at variants that are private to HG002 (*green*). Figure adapted from Reference 50.

8.1.1. Sample-specific references. Rather than implementing genotyping or variant calling over a pangenome model, the model can be used to infer a likely haploid version of a new sample's sequence, which is used as a reference genome for variant calling. These methods show incremental improvements in accuracy over conventional methods. Gramtools (87) and PanVC (139) both use specialized pangenome indexes to map reads for this purpose. Population reference graphs use read *k*-mers from reads to choose likely haplotype paths through a graph (35).

8.1.2. Small variants. In many ways, small variants stand to benefit the least from pangenomic variant calling and genotyping. NGS read lengths are sufficient to span their entirety, and the associated variant calling algorithms are quite mature. However, reference bias in mapping can be a source of small variant calling error, particularly for indels.

One strategy consists of realigning reads to graphs of known variation after mapping to a linear reference. This approach was pioneered in the 1000 Genomes Project, which applied Glia to establish genotype likelihoods for indels and complex alleles (1). GraphTyper refines this approach to achieve competitive accuracy in joint genotyping large cohorts with very low computational costs, providing improved genotyping performance at known variable sites described in its graph (40).

Colored de Bruijn graphs support pangenomic, reference-free variant calling. Cortex calls variants based on coverage in bubble structures in a colored de Bruijn graph, which is constructed from multiple read sets and/or reference genomes (65). Bubbleparse extends Cortex's model to improve SNP discovery (80).

8.1.3. Genotyping structural variation. The study of structural variants—typically defined as variants affecting at least 50 base pairs—has more to gain from using genome graphs. These variants are difficult to call with NGS reads because they are large relative to the read length. Long-read sequencing does not share this difficulty, but this technology remains prohibitively expensive for population-scale studies or routine use.

BayesTyper (128) compares the distribution of *k*-mers from sequencing reads to the distribution of *k*-mers along paths in the graph. It calls structural variants with high accuracy almost irrespective of the size of the variant. However, it has a high memory footprint, and later

analysis has also suggested that its reliance on long exact matches makes it susceptible to breakpoint uncertainty (61).

By contrast, Paragraph (26), GraphTyper2 (41), and VG call (61) use genome graphs to genotype structural variants. The largest difference is that Paragraph and GraphTyper2 first map to a linear reference, then locally realign to regional graphs, whereas VG maps reads directly to a whole-genome graph. These methods all use read coverage to determine the genotype and significantly outperform competing reference-based methods.

Some pangenomic methods have sought improved accuracy and efficiency by focusing on specific regions where reference bias makes inference challenging. The highly polymorphic and medically important human leukocyte antigen (HLA) genes have received an especially large amount of attention. HLA*LA (HLA Linear Alignments) (36), Kourami (78), and HISAT-genotype (71) have all demonstrated techniques for genotyping HLA genes by aligning NGS reads to a graph encoding various HLA alleles, and their results rival gold-standard Sanger sequencing methods in accuracy. ExpansionHunter (37) and HISAT-genotype (71) used similar methodologies to achieve comparable or better accuracy than existing methods for short tandem repeats.

8.2. Inferring Precise Haplotypes from Pangenomes and Metagenomes

Current assembly approaches often produce a result that mixes both haplotypes of a diploid genome together. This inaccurate representation has led to the development of diploid assembly methods. WHdenovo (WhatsHap de novo) (47) addresses this problem by using long reads to infer phase within the pangenomic space of the assembly graph. In an alternative approach to haplotype inference, pangenomic error correction methods can be used to clean small errors from long reads, rendering them precise observations of long haplotypes (119, 121). These methods build an assembly from accurate reads (NGS or Pacific Biosciences circular consensus), to which reads can be aligned. The path of the alignment through the graph is taken as the corrected read.

Several methods have used pangenome graphs to support haplotype reconstruction, but in the context of a mixed population of related genomes sampled from a metagenome or quasi-species mixture. Mykrobe predictor (17) and GROOT (Graphing Resistance Out of Metagenomes) (120) use graph-based structures of bacterial genomes and gene sets to predict antibody resistance in sequencing samples. MetaKallisto (122) performs taxonomic classification and quantification of metagenomic sequencing data using a database of known sequences represented as colored de Bruijn graphs. Virus-VG (10) builds a variation graph from assembled viral contigs in order to construct haplotypes and predict associated abundances in viral quasi-species from sequencing reads. This method was later improved in VG-Flow (9), which can scale to much larger genomes, such as bacteria.

8.3. Functional Pangenomics

The effects of reference bias on analyses of allele-specific protein binding and transcription have led to ample interest in pangenomic techniques. Reference bias may also affect our ability to associate genome with phenotype in association mapping. Here, we consider ways in which pangenomic models can improve the accuracy of such assays into genome function.

8.3.1. Chromatin immunoprecipitation peak calling. Chromatin immunoprecipitation sequencing data are mapped back to the reference genome in order to locate protein binding sites. Graph Peak Caller is based on VG and is the first tool to use a genome graph for this process

(55). Compared with linear methods, it was better able to find *A. thaliana* binding sites enriched for known DNA binding motifs. It was also applied to human data to discover novel sites for enhancers (54).

8.3.2. Transcriptomics. Some transcriptomic analyses are strongly affected by reference bias. Chief among these is allele-specific expression (23, 34, 133). Allele-specific expression analysis estimates the expression levels of genes or transcripts on each allele separately by comparing the number of RNA-seq reads mapped to the two different alleles of heterozygous variants. A mapping bias in favor of one of the alleles can therefore create illusory differences in expression between the alleles. Using variation information during mapping can help ameliorate this and improve estimates of allele-specific expression (23, 91).

The simplest approach to using variation data in mapping involves creating a personalized diploid genome or transcriptome, which is then used as the reference for a standard linear mapping method (113). Methods using this approach reduce reference bias but require diploid reconstruction of the genome in question. Variant-aware mappers such as GSNAP (Genomic Short-Read Nucleotide Alignment Program) (142), iMapSplice (86), ASElux (Allele-Specific Expression lux) (91), and HISAT2 (71) remove this necessity and have been shown to reduce reference bias at known single-nucleotide variants during mapping (23, 86). Variation-aware analysis of RNA-seq data is also important for accurately analyzing highly polymorphic regions, such as HLA. AltHapAlignR (79) and HLApers (3) compare reads against a collection of known HLA haplotypes, yielding improved estimation of HLA expression.

8.3.3. Pangenomic association studies. Pangenome-wide association studies generalize the genome-wide association concept to pangenomes. This new area has primarily used traditional pangenome definitions from microbiology (19). Recent work has applied pangenome-wide association studies specifically to pangenome graphs. In the frequented-region technique, a syntenic region–finding algorithm similar to those used in whole-genome alignment (Section 4.4) detects regions of a compacted de Bruijn graph that are shared among many individuals (31), then uses these regions as features in a pangenome-wide association study (89). When tested in 100 yeast strains, this approach marginally improved on standard genome-wide association study techniques, and for some phenotypes it provided a dramatic improvement in performance.

9. DISCUSSION

In the near future, we expect complete, haplotype-resolved, telomere-to-telomere assemblies of large genomes to be readily obtained at low cost (92). The impeding resolution of the genome assembly problem raises new issues: Making full use of genome assemblies will require relating them to each other, and maximizing their value will require using the prior information contained in them to guide subsequent genomic analyses.

These goals drive us to work with the pangenome implied by a collection of whole-genome assemblies. Pangenomes can be modeled as simple collections of DNA sequences, but this can obscure variation among genomes, which is essential to unlocking insight into biology. Increasingly, researchers have explored pangenome graphs that represent both sequences and variation between them. These methods are flexible: In representing the mutual alignment of many genomes, a graphical pangenome can contain and show relationships among many linear reference systems. They are also scalable: Recently developed methods support the compact storage and querying of collections of tens of thousands of genomes. And they can improve alignment and genotyping accuracy in the context of known variation.

However, adding variation to the reference system is not without potential drawbacks. Model construction, indexing, and alignment steps typically require more time for pangenome graphs than linear reference genomes. Additional information can increase ambiguity, and care must be taken to build models that improve utility by including relevant variation. Working with a graphical reference system necessitates knowledge of graph-theoretic concepts that may be unfamiliar to many biologists. Users also must consider that pangenome graphs are not observable in the same sense that a given genome is. Their construction is often guided more by application than a clear ground truth.

Due to these issues, some argue that it is likely that linear genomic models will remain important into the future (127). Our survey does not disagree with this possibility. Many of the works we have considered foresee a future in which reference systems are graphical, but only a handful (primarily those based on variation graphs) produce alignments or genotype calls in the context of a pangenome graph. Linear or hierarchical coordinate systems for the pangenome may be preferred by the genomics community. If so, these reference systems are likely to proliferate as we explore the pangenome of humans and other species.

Of course, a future full of many reference genomes is essentially a pangenomic one. Whether or not the community fosters the development of standardized pangenomic reference models, the proliferation of whole-genome sequences will only increase the importance of the methods considered here. Pangenome graphs provide a distributed framework that can be used to bring many reference systems into the same analytical context. They can also be used to build reference models optimized for particular research or clinical settings, potentially mixing public and private sources of data, without sacrificing the ability to relate the findings to standard reference models. Provided that they continue to improve, graphical pangenomic methods will be well suited to the pluralistic, decentralized attributes of a future in which genomes are easily sequenced and assembled.

DISCLOSURE STATEMENT

Multiple authors of this review are primary developers of or contributors to the VG toolkit, ODGI, GBWT, GraphAligner, seqwish, WHdenovo, and BayesTyper. J.D.S. developed prototypes based on Sequence Tube Map and is currently developing an open source visualization to compete with the tools described in this review.

AUTHOR CONTRIBUTIONS

J.M.E. wrote Sections 4.5, 6.2, 6.3, and 8.1; made **Table 2**; and helped revise the article. A.M.N. wrote Section 6.1, contributed to Section 6.3, and helped revise the article. J.A.S. wrote Sections 8.2 and 8.3 and contributed to Section 6.3. S.H. made **Table 1** and contributed to Sections 4.4 and 6.1 and **Figure 2**. A.G. made **Figure 1**. G.H. contributed to Section 8.1. X.C. contributed to Section 6.3. J.D.S. contributed to **Table 1** and contributed significantly to Section 6.1. R.R. contributed to Section 4. J.E. contributed to Section 8.1. M.R. contributed to Section 8. S.G. contributed to Section 8.2. B.P. provided guidance and opinion. T.M. contributed to Sections 1, 2, and 6.2. J.S. wrote Section 5. E.G. organized the work; wrote Sections 1–3, 7, and 9; made **Figures 2** and **3**; and helped revise the article.

ACKNOWLEDGMENTS

Research reported in this publication was supported by the National Institutes of Health under award numbers U54HG007990, U01HL137183, and 2U41HG007234. Its contents are solely

the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health. The research was also made possible by the generous financial support of the W.M. Keck Foundation (DT06172015). T.M. acknowledges funding from the German Federal Ministry for Research and Education (BMBF 031L0184). The work of J.A.S. was supported by the Carlsberg Foundation. S.H. acknowledges funding from the Central Innovation Program (ZIM) for SMEs of the Federal Ministry for Economic Affairs and Energy of Germany. J.D.S. thanks the Biotechnology and Biological Sciences Research Council for funding BB/S004661/1. We would also like to thank all the attendees of the Pangenomics Hackathon jointly organized by the National Center for Biotechnology Information and the University of California, Santa Cruz, which took place at the University of California, Santa Cruz, in the spring of 2019 and spurred many conversations that contributed to this review.

LITERATURE CITED

1. 1000 Genomes Proj. Consort. 2015. A global reference for human genetic variation. *Nature* 526:68–74
2. 1001 Genomes Consort. 2016. 1,135 genomes reveal the global pattern of polymorphism in *Arabidopsis thaliana*. *Cell* 166:481–91
3. Aguiar VRC, César J, Delaneau O, Dermitzakis ET, Meyer D. 2019. Expression estimation and eQTL mapping for HLA genes with a personalized pipeline. *PLOS Genet*. 15:e1008091
4. Ambler JM, Mulaudzi S, Mulder N. 2019. GenGraph: a python module for the simple generation and manipulation of genome graphs. *Bioinformatics* 20:519
5. Amir A, Lewenstein M, Lewenstein N. 1997. Pattern matching in hypertext. In *Algorithms and Data Structures*, ed. F Dehne, A Rau-Chaplin, JR Sack, R Tamassia, pp. 160–73. Lect. Notes Comput. Sci. 1272. Berlin: Springer
6. Antipov D, Korobeynikov A, McLean JS, Pevzner PA. 2015. hybridSPAdes: an algorithm for hybrid assembly of short and long reads. *Bioinformatics* 32:1009–15
7. Armstrong J, Hickey G, Diekhans M, Deran A, Fang Q, et al. 2019. Progressive alignment with Cactus: a multiple-genome aligner for the thousand-genome era. bioRxiv 730531. **https://doi.org/10.1101/730531**
8. Audano PA, Sulovari A, Graves-Lindsay TA, Cantsilieris S, Sorensen M, et al. 2019. Characterizing the major structural variant alleles of the human genome. *Cell* 176:663–75.e19
9. Baaijens JA, Stougie L, Schönhuth A. 2019. Strain-aware assembly of genomes from mixed samples using variation graphs. bioRxiv 645721. **https://doi.org/10.1101/645721**
10. Baaijens JA, Van der Roest B, Köster J, Stougie L, Schönhuth A. 2019. Full-length de novo viral quasispecies assembly through variation graph construction. *Bioinformatics* 35:5086–94
11. Baier U, Beller T, Ohlebusch E. 2015. Graphical pan-genome analysis with compressed suffix trees and the Burrows–Wheeler transform. *Bioinformatics* 32:497–504
12. Bernardini G, Pisanti N, Pissis SP, Rosone G. 2019. Approximate pattern matching on elastic-degenerate text. *Theor. Comput. Sci.* 812:109–22
13. Beyer W, Novak AM, Hickey G, Chan J, Tan V, et al. 2019. Sequence tube maps: making graph genomes intuitive to commuters. *Bioinformatics* 35:5318–20
14. Biederstedt E, Oliver JC, Hansen NF, Jajoo A, Dunn N, et al. 2018. NovoGraph: human genome graph construction from multiple long-read de novo assemblies. *F1000Research* 7:1391
15. Bolger A, Denton A, Bolger M, Usadel B. 2017. Logan: a framework for LOssless Graph-based ANalysis of high throughput sequence data. bioRxiv 175976. **https://doi.org/10.1101/175976**
16. Bowe A, Onodera T, Sadakane K, Shibuya T. 2012. Succinct de Bruijn graphs. In *Algorithms in Bioinformatics*, ed. B Raphael, J Tang, pp. 225–35. Lect. Notes Comput. Sci. 7534. Berlin: Springer
17. Bradley P, Gordon NC, Walker TM, Dunn L, Heys S, et al. 2015. Rapid antibiotic-resistance predictions from genome sequence data for *Staphylococcus aureus* and *Mycobacterium tuberculosis*. *Nat. Commun.* 6:10063

18. Browning SR, Browning BL. 2011. Haplotype phasing: existing methods and new developments. *Nat. Rev. Genet.* 12:703–14
19. Brynildsrud O, Bohlin J, Scheffer L, Eldholm V. 2016. Rapid scoring of genes in microbial pan-genome-wide association studies with Scoary. *Genome Biol.* 17:238
20. Büchler T, Ohlebusch E. 2019. An improved encoding of genetic variation in a Burrows-Wheeler transform. bioRxiv 658716. **https://doi.org/10.1101/658716**
21. Burrows M, Wheeler DJ. 1994. *A block sorting lossless data compression algorithm*. Tech. Rep. 124, Digital Equipment Corporation, Palo Alto, CA
22. Cao J, Schneeberger K, Ossowski S, Günther T, Bender S, et al. 2011. Whole-genome sequencing of multiple *Arabidopsis thaliana* populations. *Nat. Genet.* 43:956–63
23. Castel SE, Levy-Moonshine A, Mohammadi P, Banks E, Lappalainen T. 2015. Tools and best practices for data processing in allelic expression analysis. *Genome Biol.* 16:195
24. Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubsky D, et al. 2019. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat. Commun.* 10:1784
25. Chakraborty M, VanKuren NW, Zhao R, Zhang X, Kalsow S, Emerson J. 2018. Hidden genetic variation shapes the structure of functional elements in *Drosophila*. *Nat. Genet.* 50:20–25
26. Chen S, Krusche P, Dolzhenko E, Sherman RM, Petrovski R, et al. 2019. Paragraph: a graph-based structural variant genotyper for short-read sequence data. *Genome Biol.* 20:291
27. Chimani M, Gutwenger C, Jünger M, Klau G, Klein K, Mutzel P. 2013. The Open Graph Drawing Framework (OGDF). In *Handbook of Graph Drawing and Visualization*, ed. R Tamassia, pp. 543–69. Boca Raton, FL: CRC
28. Church DM, Schneider VA, Steinberg KM, Schatz MC, Quinlan AR, et al. 2015. Extending reference assembly models. *Genome Biol.* 16:13
29. Cisak A, Grabowski S, Holub J. 2018. SOPanG: online text searching over a pan-genome. *Bioinformatics* 34:4290–92
30. Claude F, Navarro G, Ordóñez A. 2015. The wavelet matrix: an efficient wavelet tree for large alphabets. *Inf. Syst.* 47:15–32
31. Cleary A, Ramaraj T, Kahanda I, Mudge J, Mumey B. 2018. Exploring frequented regions in pan-genomic graphs. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 16:1424–35
32. Comput. Pan-Genom. Consort. 2016. Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* 19:118–35
33. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, et al. 2011. The variant call format and VCFtools. *Bioinformatics* 27:2156–58
34. Degner JF, Marioni JC, Pai AA, Pickrell JK, Nkadori E, et al. 2009. Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. *Bioinformatics* 25:3207–12
35. Dilthey AT, Cox C, Iqbal Z, Nelson MR, McVean G. 2015. Improved genome inference in the MHC using a population reference graph. *Nat. Genet.* 47:682–88
36. Dilthey AT, Mentzer AJ, Carapito R, Cutland C, Cereb N, et al. 2019. HLA*LA—HLA typing from linearly projected graph alignments. *Bioinformatics* 35:4394–96
37. Dolzhenko E, Deshpande V, Schlesinger F, Krusche P, Petrovski R, et al. 2019. ExpansionHunter: a sequence-graph-based tool to analyze variation in short tandem repeat regions. *Bioinformatics* 35:4754–56
38. Duan Z, Qiao Y, Lu J, Lu H, Zhang W, et al. 2019. HUPAN: a pan-genome analysis pipeline for human genomes. *Genome Biol.* 20:149
39. Durbin R. 2014. Efficient haplotype matching and storage using the positional Burrows-Wheeler transform (PBWT). *Bioinformatics* 30:1266–72
40. Eggertsson HP, Jonsson H, Kristmundsdottir S, Hjartarson E, Kehr B, et al. 2017. Graphtyper enables population-scale genotyping using pangenome graphs. *Nat. Genet.* 49:1654–60
41. Eggertsson HP, Kristmundsdottir S, Beyter D, Jonsson H, Skuladottir A, et al. 2019. GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nat. Commun.* 10:5402
42. Ferragina P, Manzini G. 2005. Indexing compressed text. *J. ACM* 52:552–81

43. Franz M, Lopes C, Huck G, Dong Y, Sumer O, Bader G. 2016. Cytoscape.js: a graph theory library for visualization and analysis. *Bioinformatics* 32:309–11
44. Fu S, Wang A, Au KF. 2019. A comparative evaluation of hybrid error correction methods for error-prone long reads. *Genome Biol*. 20:26
45. Gagie T, Manzini G, Sirén J. 2017. Wheeler graphs: a framework for BWT-based data structures. *Theor. Comput. Sci.* 698:67–78
46. Gao L, Gonda I, Sun H, Ma Q, Bao K, et al. 2019. The tomato pan-genome uncovers new genes and a rare allele regulating fruit flavor. *Nat. Genet.* 51:1044–51
47. Garg S, Rautiainen M, Novak AM, Garrison E, Durbin R, Marschall T. 2018. A graph-based approach to diploid genome assembly. *Bioinformatics* 34:i105–14
48. Garrison E. 2019. *Graphical pangenomics*. PhD Thesis, Univ. Cambridge, Cambridge, UK
49. Garrison E, Marth G. 2012. Haplotype-based variant detection from short-read sequencing. arXiv:1207.3907 [q-bio.GN]
50. Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, et al. 2018. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* 36:875–79
51. Ghaffaari A, Marschall T. 2019. Fully-sensitive seed finding in sequence graphs using a hybrid index. *Bioinformatics* 35:i81–89
52. Gonnella G, Niehus N, Kurtz S. 2018. GfaViz: flexible and interactive visualization of GFA sequence graphs. *Bioinformatics* 35:2853–55
53. Grasso C, Lee C. 2004. Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics* 20:1546–56
54. Groza C, Kwan T, Soranzo N, Pastinen T, Bourque G. 2019. Personalized and graph genomes reveal missing signal in epigenomic data. bioRxiv 457101. **https://doi.org/10.1101/457101**
55. Grytten I, Rand KD, Nederbragt AJ, Storvik GO, Glad IK, Sandve GK. 2019. Graph Peak Caller: calling ChIP-seq peaks on graph-based reference genomes. *PLOS Comput. Biol.* 15:e1006731
56. Haeussler M, Zweig AS, Tyner C, Speir ML, Rosenbloom KR, et al. 2018. The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res*. 47:D853–58
57. Hehir-Kwa JY, Marschall T, Kloosterman WP, Francioli LC, Baaijens JA, et al. 2016. A high-quality human reference panel reveals the complexity and distribution of genomic structural variants. *Nat. Commun.* 7:12989
58. Hein J. 1989. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. *Mol. Biol. Evol.* 6:649–68
59. Heng L, Handsaker B, Wysoker A, Fennell T, Ruan J, et al. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25:2078–79
60. Heydari M, Miclotte G, Van de Peer Y, Fostier J. 2018. BrownieAligner: accurate alignment of Illumina sequencing data to de Bruijn graphs. *BMC Bioinform*. 19:311
61. Hickey G, Heller D, Monlong J, Sibbesen JA, Sirén J, et al. 2019. Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome Biol*. 21:35
62. Holley G, Melsted P. 2019. Bifrost – highly parallel construction and indexing of colored and compacted de Bruijn graphs. bioRxiv 695338. **https://doi.org/10.1101/695338**
63. Huang L, Popic V, Batzoglou S. 2013. Short read alignment with populations of genomes. *Bioinformatics* 29:i361–70
64. Huang S, Lam T, Sung W, Tam S, Yiu S. 2010. Indexing similar DNA sequences. In *Algorithmic Aspects in Information and Management*, ed. B Chen, pp. 180–90. Lect. Notes Comput. Sci. 6124. Berlin: Springer
65. Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. 2012. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.* 44:226–32
66. Jain C, Misra S, Zhang H, Dilthey A, Aluru S. 2019. Accelerating sequence alignment to graphs. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 451–61. Piscataway, NJ: IEEE
67. Jain C, Zhang H, Gao Y, Aluru S. 2019. On the complexity of sequence to graph alignment. In *Research in Computational Molecular Biology*, ed. L Cowen, pp. 85–100. Cham, Switz.: Springer
68. Jandrasits C, Dabrowski PW, Fuchs S, Renard BY. 2018. seq-seq-pan: building a computational pan-genome data structure on whole genome alignment. *BMC Genom*. 19:47

69. Kavya VNS, Tayal K, Srinivasan R, Sivadasan N. 2019. Sequence alignment on directed graphs. *J. Comput. Biol.* 26:53–67
70. Kim D, Langmead B, Salzberg SL. 2015. HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods* 12:357–60
71. Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. 2019. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat. Biotechnol.* 37:907–15
72. Krzywinski M, Schein J, Birol I, Connors J, Gascoyne R, et al. 2009. Circos: an information aesthetic for comparative genomics. *Genome Res.* 19:1639–45
73. Kunyavskaya O, Prjibelski AD. 2018. SGTK: a toolkit for visualization and assessment of scaffold graphs. *Bioinformatics* 35:2303–5
74. Kural D. 2014. *Methods for inter-and intra-species genomics for the detection of variation and function*. PhD Thesis, Boston Coll., Boston
75. Laing C, Buchanan C, Taboada EN, Zhang Y, Kropinski A, et al. 2010. Pan-genome sequence analysis using Panseq: an online tool for the rapid analysis of core and accessory genomic regions. *BMC Bioinform.* 11:461
76. Langley SA, Miga KH, Karpen GH, Langley CH. 2019. Haplotypes spanning centromeric regions reveal persistence of large blocks of archaic DNA. *eLife* 8:e42989
77. Lee C, Grasso C, Sharlow MF. 2002. Multiple sequence alignment using partial order graphs. *Bioinformatics* 18:452–64
78. Lee H, Kingsford C. 2018. Kourami: graph-guided assembly for novel human leukocyte antigen allele discovery. *Genome Biol.* 19:16
79. Lee W, Plant K, Humburg P, Knight JC. 2018. AltHapAlignR: improved accuracy of RNA-seq analyses through the use of alternative haplotypes. *Bioinformatics* 34:2401–8
80. Leggett RM, Ramirez-Gonzalez RH, Verweij W, Kawashima CG, Iqbal Z, et al. 2013. Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de Bruijn graphs. *PLOS ONE* 8:e60058
81. Li H. 2011. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* 27:2987–93
82. Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio.GN]
83. Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 34:3094–100
84. Li R, Li Y, Zheng H, Luo R, Zhu H, et al. 2010. Building the sequence map of the human pan-genome. *Nat. Biotechnol.* 28:57–63
85. Linthorst J, Hulsman M, Holstege H, Reinders M. 2015. Scalable multi whole-genome alignment using recursive exact matching. bioRxiv 022715. **https://doi.org/10.1101/022715**
86. Liu X, MacLeod JN, Liu J. 2018. iMapSplice: alleviating reference bias through personalized RNA-seq alignment. *PLOS ONE* 13:e0201554
87. Maciuca S, del Ojo Elias C, McVean G, Iqbal Z. 2016. A natural encoding of genetic variation in a Burrows-Wheeler transform to enable mapping and genome inference. In *Algorithms in Bioinformatics*, ed. M Frith, CNS Pedersen, pp. 222–33. Lect. Notes Comput. Sci. 9838. Cham, Switz.: Springer
88. Mäkinen V, Navarro G, Sirén J, Välimäki N. 2010. Storage and retrieval of highly repetitive sequence collections. *J. Comput. Biol.* 17:281–308
89. Manuweera B, Mudge J, Kahanda I, Mumey B, Ramaraj T, Cleary A. 2019. Pangenome-wide association studies with frequented regions. In *ACM-BCB'19: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 627–32. New York: ACM
90. Marcus S, Lee H, Schatz MC. 2014. SplitMEM: a graphical algorithm for pan-genome analysis with suffix skips. *Bioinformatics* 30:3476–83
91. Miao Z, Alvarez M, Pajukanta P, Ko A. 2018. ASElux: an ultra-fast and accurate allelic reads counter. *Bioinformatics* 34:1313–20
92. Miga KH, Koren S, Rhie A, Vollger MR, Gershman A, et al. 2019. Telomere-to-telomere assembly of a complete human X chromosome. bioRxiv 735928. **https://doi.org/10.1101/735928**
93. Mikheenko A, Kolmogorov M. 2019. Assembly Graph Browser: interactive visualization of assembly graphs. *Bioinformatics* 35:3476–78

94. Minkin I, Medvedev P. 2019. Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. bioRxiv 548123. **https://doi.org/10.1101/548123**
95. Minkin I, Pham S, Medvedev P. 2016. TwoPaCo: an efficient algorithm to build the compacted de Bruijn graph from many complete genomes. *Bioinformatics* 33:4024–32
96. Mokveld TO, Linthorst J, Al-Ars Z, Reinders M. 2018. CHOP: haplotype-aware path indexing in population graphs. bioRxiv 305268. **https://doi.org/10.1101/305268**
97. Myers EW. 2005. The fragment assembly string graph. *Bioinformatics* 21:ii79–85
98. Myers EW, Miller W. 1989. Approximate matching of regular expressions. *Bull. Math. Biol.* 51:5–37
99. Na JC, Kim H, Min S, Park H, Lecroq T, et al. 2018. FM-index of alignment with gaps. *Theor. Comput. Sci.* 710:148–57
100. Na JC, Kim H, Park H, Lecroq T, Léonard M, et al. 2016. FM-index of alignment: a compressed index for similar strings. *Theor. Comput. Sci.* 638:159–70
101. Navarro G. 2000. Improved approximate pattern matching on hypertext. *Theor. Comput. Sci.* 237:455–63
102. Notredame C, Higgins DG, Heringa J. 2000. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.* 302:205–17
103. Novak AM, Garrison E, Paten B. 2017. A graph extension of the positional Burrows–Wheeler transform and its applications. *Algorithms Mol. Biol.* 12:18
104. Novak AM, Hickey G, Garrison E, Blum S, Connelly A, et al. 2017. Genome graphs. bioRxiv 101378. **https://doi.org/10.1101/101378**
105. Onodera T, Sadakane K, Shibuya T. 2013. Detecting superbubbles in assembly graphs. In *Algorithms in Bioinformatics*, ed. A Darling, J Stoye, pp. 338–48. Lect. Notes Comput. Sci. 8126. Berlin: Springer
106. Ou L, Li D, Lv J, Chen W, Zhang Z, et al. 2018. Pan-genome of cultivated pepper (*Capsicum*) and its use in gene presence-absence variation analyses. *New Phytol.* 220:360–63
107. Page AJ, Cummins CA, Hunt M, Wong VK, Reuter S, et al. 2015. Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics* 31:3691–93
108. Paten B, Eizenga JM, Rosen YM, Novak AM, Garrison E, Hickey G. 2018. Superbubbles, ultrabubbles, and cacti. *J. Comput. Biol.* 25:649–63
109. Paten B, Novak AM, Eizenga JM, Garrison E. 2017. Genome graphs and the evolution of genome inference. *Genome Res.* 27:665–76
110. Pevzner PA, Tang H, Waterman MS. 2001. An Eulerian path approach to DNA fragment assembly. *PNAS* 98:9748–53
111. Poplin R, Ruano-Rubio V, DePristo MA, Fennell TJ, Carneiro MO, et al. 2018. Scaling accurate genetic variant discovery to tens of thousands of samples. bioRxiv 201178. **https://doi.org/10.1101/201178**
112. Pritt J, Chen NC, Langmead B. 2018. FORGe: prioritizing variants for graph genomes. *Genome Biol.* 19:220
113. Raghupathy N, Choi K, Vincent MJ, Beane GL, Sheppard KS, et al. 2018. Hierarchical analysis of RNA-seq reads improves the accuracy of allele-specific expression. *Bioinformatics* 34:2177–84
114. Rahn R, Weese D, Reinert K. 2014. Journaled string tree—a scalable data structure for analyzing thousands of similar genomes on your laptop. *Bioinformatics* 30:3499–505
115. Rakocevic G, Semenyuk V, Lee WP, Spencer J, Browning J, et al. 2019. Fast and accurate genomic analyses using genome graphs. *Nat. Genet.* 51:354–62
116. Rand KD, Grytten I, Nederbragt AJ, Storvik GO, Glad IK, Sandve GK. 2017. Coordinates and intervals in graph-based reference genomes. *BMC Bioinform.* 18:263
117. Rautiainen M, Mäkinen V, Marschall T. 2019. Bit-parallel sequence-to-graph alignment. *Bioinformatics* 35:3599–607
118. Rautiainen M, Marschall T. 2017. Aligning sequences to general graphs in $O(V + mE)$ time. bioRxiv 216127. **https://doi.org/10.1101/216127**
119. Rautiainen M, Marschall T. 2019. GraphAligner: rapid and versatile sequence-to-graph alignment. bioRxiv 810812. **https://doi.org/10.1101/810812**
120. Rowe WPM, Winn MD. 2018. Indexed variation graphs for efficient and accurate resistome profiling. *Bioinformatics* 34:3601–8
121. Salmela L, Walve R, Rivals E, Ukkonen E. 2016. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics* 33:799–806

122. Schaeffer L, Pimentel H, Bray N, Melsted P, Pachter L. 2017. Pseudoalignment for metagenomic read assignment. *Bioinformatics* 33:2082–88
123. Schmidt D, Colomb R. 2009. A data structure for representing multi-version texts online. *Int. J. Hum.-Comput. Stud.* 67:497–514
124. Schneeberger K, Hagmann J, Ossowski S, Warthmann N, Gesing S, et al. 2009. Simultaneous alignment of short reads against multiple genomes. *Genome Biol*. 10:R98
125. Sheikhizadeh Anari S, de Ridder D, Schranz ME, Smit S. 2018. Efficient inference of homologs in large eukaryotic pan-proteomes. *BMC Bioinform*. 19:340
126. Sherman RM, Forman J, Antonescu V, Puiu D, Daya M, et al. 2019. Assembly of a pan-genome from deep sequencing of 910 humans of African descent. *Nat. Genet.* 51:30–35
127. Sherman RM, Salzberg SL. 2020. Pan-genomics in the human genome era. *Nat. Rev. Genet.* 21:243–54
128. Sibbesen JA, Maretty L, Dan. Pan-Genome Consort, Krogh A. 2018. Accurate genotyping across variant classes and lengths using variant graphs. *Nat. Genet*. 50:1054–59
129. Sirén J. 2017. Indexing variation graphs. In *Proceedings of the Nineteenth Meeting on Algorithm Engineering and Experiments (ALENEX 2017)*, ed. S Fekete, V Ramachandran, pp. 13–27. Philadelphia: Soc. Ind. Appl. Math.
130. Sirén J, Garrison E, Novak AM, Paten B, Durbin R. 2020. Haplotype-aware graph indexes. *Bioinformatics* 36:400–7
131. Sirén J, Välimäki N, Mäkinen V. 2014. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 11:375–88
132. Smith TF, Waterman MS. 1981. Comparison of biosequences. *Adv. Appl. Math.* 2:482–89
133. Stevenson KR, Coolon JD, Wittkopp PJ. 2013. Sources of bias in measures of allele-specific expression derived from RNA-seq data aligned to a single reference genome. *BMC Genom*. 14:536
134. Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, et al. 2015. An integrated map of structural variation in 2,504 human genomes. *Nature* 526:75–81
135. Suzuki H. 2018. Dozeu. *GitHub*. **https://github.com/ocxtal/dozeu**
136. Thachuk C. 2013. Indexing hypertext. *J. Discrete Algorithms* 18:113–22
137. Turner I, Garimella KV, Iqbal Z, McVean G. 2018. Integrating long-range connectivity information into de Bruijn graphs. *Bioinformatics* 34:2556–65
138. Vaddadi K, Srinivasan R, Sivadasan N. 2019. Read mapping on genome variation graphs. In *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*, ed. KT Huber, D Gusfield, art. 7. Dagstuhl, Ger.: Schloss Dagstuhl–Leibniz-Zent. Inform.
139. Valenzuela D, Norri T, Välimäki N, Pitkänen E, Mäkinen V. 2018. Towards pan-genome read alignment to improve variation calling. *BMC Genom*. 19:87
140. Vernikos G, Medini D, Riley DR, Tettelin H. 2015. Ten years of pan-genome analyses. *Curr. Opin. Microbiol.* 23:148–54
141. Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* 31:3350–52
142. Wu TD, Nacu S. 2010. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics* 26:873–81
143. Yokoyama TT, Sakamoto Y, Seki M, Suzuki Y, Kasahara M. 2019. MoMI-G: modular multi-scale integrated genome graph browser. *BMC Bioinform*. 20:548
144. Yue JX, Li J, Aigrain L, Hallin J, Persson K, et al. 2017. Contrasting evolutionary genome dynamics between domesticated and wild yeasts. *Nat. Genet.* 49:913–24
145. Zhao Y, Wu J, Yang J, Sun S, Xiao J, Yu J. 2011. PGAP: pan-genomes analysis pipeline. *Bioinformatics* 28:416–18
146. Zhou B, Wen S, Wang L, Jin L, Li H, Zhang H. 2017. AntCaller: an accurate variant caller incorporating ancient DNA damage. *Mol. Genet. Genom.* 292:1419–30
147. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, et al. 2014. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* 32:246–51

Annual Review of Genomics and Human Genetics

Volume 21, 2020

Contents