J. Indian Inst. Sci.
A Multidisciplinary Reviews Journal
ISSN: 0970-4140 Coden-JIISAD

REVIEW ARTICLE

Methods and Developments in Graphical Pangenomics

*Joseph Outten and Andrew Warren**

Abstract | Pangenomes are organized collections of the genomic information from related individuals or groups. Graphical pangenomics is the study of these pangenomes using graphical methods to identify and analyze genes, regions, and mutations of interest to an array of biological questions. This field has seen significant progress in recent years including the development of graph based models that better resolve biological phenomena, and an explosion of new tools for mapping reads, creating graphical genomes, and performing pangenome analysis. In this review, we discuss recent developments in models, algorithms associated with graphical genomes, and comparisons between similar tools. In addition we briefly discuss what these developments may mean for the future of genomics.

Keywords: *Multiple sequence alignment, Genome assembly, Graph genomes, Pangenomics, Graphical pangenomics*

1 Introduction

Pangenomes were first introduced in 2005 by Tettelin et al.[69] in the context of microbial populations. This early conception of pangenomes consists of a core genome and a dispensable genome. The core genome includes the sequence features, usually genes, which are shared by all individuals in the analyzed group (such as a species or genus), while the accessory genome includes all other genes which are only partially shared among the group. Canonical representations of the conservation and diversity present in a group of genomes can be given through clustering, single nucleotide polymorphisms (SNP), phylogenetic trees, and multiple sequence alignments[71]. The analysis of these similarities and differences between related genomes is the subject of many studies investigating whole genes, or specific mutations, that may have some significance, clinical, biological, or otherwise.

Since the initial formulation of pangenomes as a collection of sequences, much work has been done to define relevant data structures, algorithms, and applications. This work has included different tools to represent and operate over genome sets, with applications to repetitive regions, cancer genomes, variant calling, genotyping, evolutionary analysis, and other topics[10,38,72]. In recent years, graphical models have come to the forefront of pangenomic analysis, especially for human genomes[60]. While the specific instantiation of these graphical models can vary considerably, many modern formulations use a sequence graph, where nodes correspond to segments of the genome (sequence strings) and edges connect adjacent segments in a directed or bidirected manner.

Graph theorists may be familiar with the concept of a de Bruijn graph, which represents similarities in sequences of symbols according to a parameter k such that nodes represent a region of length k matching symbols from two or more strings. Edges represent a $k-1$ overlap of matching symbols between a pair of nodes. Similarly sequence graphs represent matched sequence segments from multiple strings as nodes. The sequence graph takes on the double stranded nature of DNA by labelling nodes with both the forward and reverse complement of the DNA string. This accommodation results in the sequence graph model being necessarily bidirected. Modern algorithms that build pangenome graph models make use of sequence graphs to represent the relationships among different sets

Conserved sequences: Sequences of DNA or proteins that are maintained across species or individuals through evolutionary processes.

SNP: Differences between sets of DNA sequences occurring at a singl position.

Variant calling: The process of determining differences, called variants, between a query genome and some reference genome(s).

Genotyping:: The process of determining the specific genetic sequence of an individual at any or all locations in their genome.

[1] *Biocomplexity Institute, University of Virginia, Charlottesville, USA.*
**anwarren@virginia.edu*

of strings, i.e. genomes, and apply different mapping criteria to establish a frame of reference for a matched sequence or subsequence.

Graphical genomes seek to reduce reference bias for population level analysis. Reference bias occurs when a single reference sequence (or a limited set of reference sequences) is used as a common guide to interpret many other similar genomes, leading to biased perspectives or analysis. This can be seen in the identification and analysis of SNPs for the human genome. When a new genome is sequenced, it is usually done in short strings that are stitched together through the creation and transformation of a de Bruijn graph, a process known as assembly. The de novo assembly process commonly involves mapping the short strings, reads, to one another but if a genome is re-sequenced for comparative purposes, as is commonly the case for SARS-CoV-2, the reads are aligned to a reference. If the new genome has some region that diverges significantly from the reference, it is frequently ignored in downstream analysis. Graphical genomes allow the explicit and compact representation of the diversity in sequence found in many samples, helping to reduce this bias. The shift away from linear reference analysis has blurred the lines between what constitutes genomic and pangenomic analysis.

As sequencing efforts and variant catalogues advance to categorize the variation present in human and non-human populations, significant differences between populations and established reference sequences have become better resolved. Graphical representations of pangenomes are powerful due to an explicit capturing of similarities and differences in the node-edge modality and an ability to define nested variation which can lead to advantages in topological analysis and reduced bias compared to a reference-centric model. Several recent reviews have covered these topics well[10,19,52]. In this paper, we summarize new tools and developments for modeling pangenomes as graphs, and discuss proposed formats as standards for graphical pangenome analysis as well as remaining challenges and limitations in this growing field. A list of terms relevant to this discussion is given in Table 1.

Contig: A stretch of sequence (usually DNA) which is continuous in the genome. Usually resolved from overlapping short reads produced from DNA sequencing.

2 Model Fundamentals

2.1 Similarity Context

Comparing multiple, usually divergent, string sequences of DNA in a pangenome requires the determination of comparative coordinate groups, using a frame of reference to establish the similarities and differences between each string. Many graphical pangenome methods create a graph model of a global, multiple sequence alignment, with potentially varying objective functions and alphabets. As of this writing, many recent methods for graphical pangenome analysis instantiate a genome graph. There are some variations to this concept in the literature. Here we adhere to the hierarchy of graph types defined by Paten et al. [52]. A genome graph uses the DNA alphabet to express similarity among contigs given as constituent strings of the input genomes. All genome graph creation algorithms, establish multiple sets of string coordinates as either explicit or implicit labels on the resulting nodes, we call an instance of this set a *similarity group*.

In order to determine which intervals on which strings to compare, most algorithms that create a pangenome graph use a *mapping criteria* to establish a frame of reference, which can be thought of as a set of intervals on the input strings bounded and grouped by a fulfillment of the mapping criteria. The resulting frame of reference is then evaluated, and potentially discarded or deconvoluted into one or more similarity groups. Here similarity groups can be thought of as a set function that glues coordinates on different strings together according to the frame of reference established by the mapping criteria. Current methods typically express the labels, derived from the frame of reference, relative to a curated input genome designated a "reference genome"[35] or relative to the graph output[51]. For the majority of new methods the criteria by which the frame of reference is evaluated is not explicitly related to a global optimum[15,26,47]. Instead the similarity groupings, and by implication the mapping criteria, are determined by the type of similarities the graph model is intended to embody and the manifest differences it is able to detect[49].

In graphical pangenomics, the individual input sequences can be represented as a walk through the graph structure. What properties of the similarities and differences among the input strings are defined depend on the detail of the graph used. When those sequences are matched, according to the mapping criteria and deconvolution logic, they are represented as coincident labels on nodes and edges. Noting that both nodes and edges are capable of being labelled with sequences, for this review we subscribe to the convention that nodes are labelled with sequence intervals and edges represent a

Table 1: Terminology, informed in part by[19,52].

Term	Description	Parent types*
de Bruijn graph	Nodes are sequence k–mers, and directed edges connect k–mers whose k-1 suffix overlaps with other k–mers k-1 prefix	NA
Sequence graph	Edges or nodes are labelled with sequences. Used to compress sequence representation and express contiguity between segments with directed or bidirected edges	NA
Genome graph	Relates a genome's sequence information to itself or other genomes	Sequence graph
Pangenome	A representation of the genetic information across a population	NA
Pangenome graph	Genome graphs explicitly involving more than one genome	Genome graph, sequence graph
Synteny graph	Relates blocks of conserved sequence	Sequence graph
Reference genome	Used as the standard for comparison in a species, e.g. GRCh38	NA
Reference bias	The use of a linear-reference causing incomplete analysis or a lack of sensitivity	NA
Variation graph	Bidirected graphs which embed linear sequences as paths	Pangenome graph, bidirected graph
Bidirected graph	Each edge has a discrete endpoint on either the left or right of a node	Sequence graph

*Where applicable, parent types lists those other terms for which the term in question is a specific type

contiguity relationship for those intervals whose sequences span them.

Pangenome graph creation often employs hashing to form the basis of the mapping criteria wherein the exact match in question is required to be of at least a defined length k as seen in Minkin et al.[46]. These methods usually create a de Bruijn graph (see Fig. 1), either explicitly or implicitly[42], which are then used to resolve the desired relationships in the model depending on the application. Though de Bruijn graph based methods typically apply a k–mer exact match criteria ubiquitously, in principle there is no reason the frame of reference cannot be generated by a range of mapping criteria such as regular expressions, spectral clustering, and other more variant tolerant approaches, e.g. Dilthey et al.[12] uses Hidden Markov Models (HMM's) to establish the mapping criteria. Depending on the application, graph creation may apply different requirements on the mapping criteria and frame of reference. There is a need in this area to formalize the guarantees and implications associated with different mapping criteria and the model resolution and subsequent interpretation. Currently such implications are usually given as constraints on the type of output or model generated, e.g. discovering structural variants (SVs) larger than 100 base pair in Heng Li's minigraph[34]. This variation in output model, forms the basis of our resolution discussion below.

Genome graphs and their intermediates are used in various contexts including assembly, metage- nome assembly, and SNP calling[66,73]. In these contexts what varies are the labels applied to the graph and the constraints applied to deconvoluting intermediate forms to arrive at a final result. Labels applied to the graph for these transformations typically stem from, potentially putative, sample, organism, contig, or replicon information. The grouping of sequence intervals on the basis of mapping criteria is sometimes referred to as "glue", and is often used to deconvolute the pangenomic model based on the labels involved. Iqbal et al.[27] conceptualize labels for samples as a colored de Bruijn graph for determining SNPs. Separating these labels based on the desired outcome is often the basis of creating similarity groups. Taking the example from Fig. 1 panel i, if SNP calling is the objective among the three sequences then the mapping criteria either accommodates gaps or the frame of reference is established such that similarity groups are created at single character resolution with a guarantee that the mapped sequences minimize ambiguity according to other putative homologous relationships. Larger variants, Fig. 1 panel ii, can be difficult to detect in a traditional mapping based workflows depending on the length of the sequencing read. In both cases reference bias can confound the detection of variation, especially in regions of high diversity.

Structural Variant: A DNA variant usually longer than 50 letters. Can be inversions, deletions, duplications, etc.

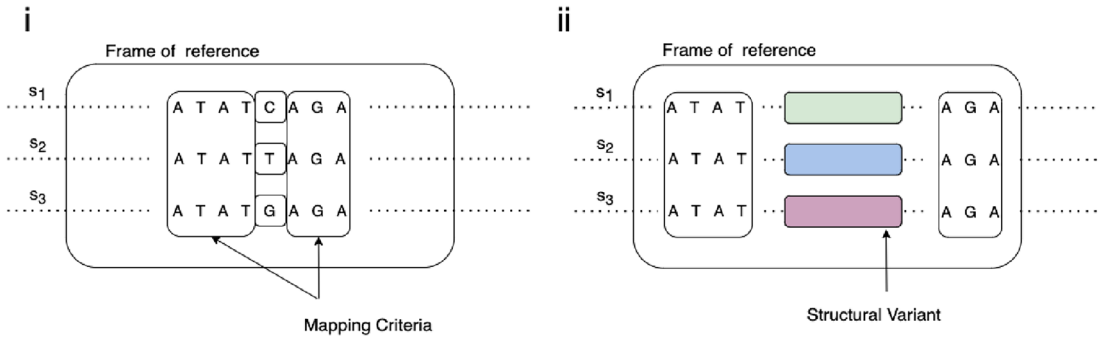

Figure 1: (i) An example of a genome graph at the resolution of single nucleotide polymorphisms. The mapping criteria of exact match $k = 3$, is used to define a frame of reference and the resulting nodes (similarity groups). (ii) An example of a larger structural variant. The colored bars represent larger graph structures which themselves represent divergent sequences that do not meet the mapping criteria relative to one another.

Genome graph disambiguation is subject to provenance information concerning contigs and their origin. Figure 2 gives a contrived example to demonstrate the limit of disambiguation for metagenomic and pangenomic graph construction. Applying the label disambiguation concept to pangenome graph creation in a metagenome context, where the intermediate of a genome graph is often represented as a de Bruijn graph (Fig. 2, panel ii), has the limitation that the source genome for a given contig is unknown. In Fig. 2 panel iii we see that the best resolution for a genome graph in this context. Though two pairs of contigs come from the same source genome, without reference information or sample provenance given from a clonal sampling, the limit of disambiguation is to form similarity groups which include multiple intervals from the same genome. When genome graph creation is given assembled genomes, the source of all contigs are known and can employ additional disambiguation to ensure that duplicated segments both within the same contig and within the same genome are not resolved to the same similarity group, Fig. 2 panel iv. The consequences of repeat regions and methods that address this topic are highlighted below.

Metagenome: The complete genomic content contained in an environ-mental sample, potentiallyfrom mny different organisms

2.2 Regularized Differences

Within a given frame of reference there may be sub-intervals in sequences that are divergent from one another. In graph genome models, assembly, SNP calling, and metagenome assembly these subintervals are represented by bubbles[75]. Bubbles represent two paths that are disjoint outside a defined sink and source point representing where the divergence begins and ends. Bubbles can be created by single nucleotide polymorphisms, and other larger SVs. Paten et al.[51] defines a hierarchy of applicable bubble types in their discussion of superbubbles and ultrabubbles as they apply to bidirected graphs. In short, superbubbles expand on the notion of a bubble by removing the condition that the paths be disjoint but still result in a subgraph of a bidirected graph with an existing sink and source node. Inversions and translocations at the genome scale often produce superbubbles. Most recent graphical pangenome tools[17,23,35,58] represent sequence graphs using bidirected graphs or their equivalent. This is due to the bidirected graph being able to better capture inversions, duplications, and other complex rearrangements[52].

2.3 Repeat Regions

Repeats are segments of DNA found multiple times throughout a genome. These regions pose significant problems for assembly, mapping, alignment, and genotyping algorithms since there is often ambiguity in where each repeat belongs in the linearized genome and where reads containing these repeats should map to, especially short reads which can be shorter than the repeat sequence. Since they are hypervariable in number and location, any given assembly (such as the current human linear reference) is a poor representation of repeats in the population[38]. It is common to mask away these repeat regions before analysis of whole genome sequencing data, but several studies have shown their biological importance[4,44]. As such, Slotkin et al.[67] argued against this masking step and noted that 25 times

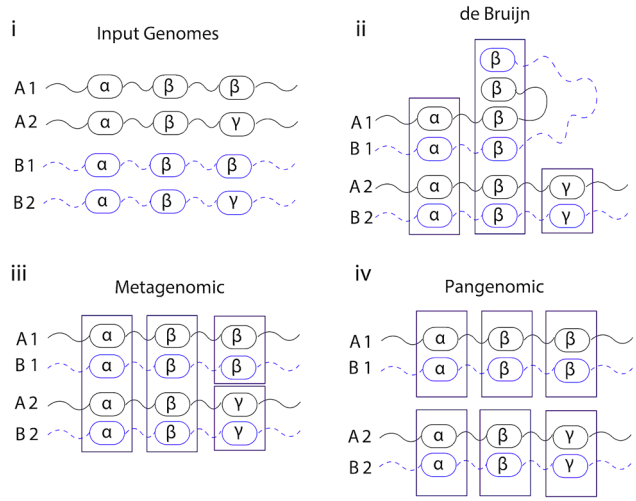

Figure 2: Examples of deconvolution. Regions of similarity have matching symbols and connecting edges, and by logical extension diverging paths represent regions of divergence. Boxes represent a similarity group, which forms the basis of including a region in a node for the multi-genome model. The extent of deconvolution is dictated by the incoming labels, the mapping criteria, and the frame of reference given by the algorithm. (i) Two input genomes A and B each with two replicons. (ii) A de Bruijn graph model of similarity is dictated by *k*-mer parameter size and the amount of repeat similarity. (iii) An example of a metagenomic level of resolution capable in a genome graph given unknown provenance of contigs. (iv) Fully disambiguated groupings for assembled genomes given as input.

as much sequence is discovered in studies which consider repeats than those which ignored them. Repeats pose a challenge to traditional methods, but must be included in analyses to get a complete picture of the relationship between genotype and phenotype.

While they do not solve all the issues associated with repeats, methods using graphical pangenomes have shown the potential to effectively handle these and other complex variants. Some approaches have seen a decrease in memory needed to represent pangenomes by collapsing repeats into a single node. This is done as part of de Bruijn graph construction and can be incorporated into sequence and variation graphs by allowing cycles[21,29]. Other tools for genotyping SVs based on graphical genomes have shown improvements over traditional methods, but genotyping using SVs composed of repeats remains a challenging disambiguation task [7,23].

A handful of approaches have been designed explicitly to handle repeats using a graph based approach, including ExpansionHunter[13] for short tandem repeats (STRs) and the danbing-tk toolkit[38] for variable number tandem repeats (VNTRs). Both tools use locus specific models built on known variation for genotyping; ExpansionHunter uses a sequence graph implementation while the danbing-tk toolkit uses de Bruijn graphs. These examples show that graphical, pangenomic methods, may allow more accurate analyses and complete representation of the "repeatome" as discussed in[52]. A potential limitation is that these methods are based known sources of variation and thus may suffer from their own reference bias inherent to those sources. Advances in long read sequencing technologies, in tandem with graphical pangenomic methods, are poised to provide significant improvements in the study of repeats.

2.4 Haplotypes

Haplotypes, in the context of graphical pangenomes, are paths. Each path through a de Bruijn, sequence, overlap, variation, or other graph, is a potential haplotype. However, linkage disequilibrium, in its most basic form, causes loci which are closer together to be inherited together at a higher rate. In this way, some haplotypes, or paths through the graph, are more likely to exist than others. This means that known haplotypes are valuable in both their content and order of composition.

Several implementations of pangenome graphs implicitly take advantage of known haplotypes, including colored de Bruijn graphs[36] and methods which build sample, or dataset, specific

Locus: The specific region of a genome containing a gene or sequence of interest.

references[9,40,70]. Compacted colored de Bruijn graphs (ccDBGs) are de Bruijn graphs which have had non-branching edges collapsed (compacted) and have each node (or sequence segment) colored by which genomes or samples it appears in. Several methods have recently been developed to efficiently build, store, and query these graphs[3,24,28,42,46] and even use them for read alignment[27,31,36]. However concerns have been raised over the ability of de Bruijn graph based tools for representing large repetitive elements and for scaling to high cardinality sets of large mammalian genomes[19,34].

Syntenic Region: Regions of co-localized loci or genes in a genome or pangenome.

Unlike the previous examples, pangenomes encoded in variation graphs do not natively encode or take advantage of haplotype information. Several tools for compressing haplotype path indexes through variation graphs exist, including gPBWT[48], the graph extension of the positional Borrows-Wheeler transform (PBWT)[14] and GBWT[65], the graph extension of the Burrows-Wheeler transform[5], provided by the vg toolkit[66]. These methods support efficient searching and matching queries.

There is some variation in how the powerful prior of haplotypes are employed across tools for graphical pangenomics. Since the number of possible paths is exponential in the amount of variation (both locally and globally), many implementations of graphical genomes "prune" the graph to allow indexing in reasonable resource constraints (although alignment may be done to the full graph)[19]. Therefore, one use of known haplotypes is to help identify which paths should be indexed and which can be thrown out, leading to a haplotype-aware indexing strategy[65], as used in the vg toolkit. Other tools use haplotypes to build probabilistic models for genotyping SVs, indels, and SNPs[7,66], or simply do not incorporate them into their indexes or analyses[17,34,56].

Indel: Insertions and deletions of sequence segments in a genome.

While observed haplotypes are powerful resources for building effective and efficient tools for graphical pangenomes, their complement also carries potentially valuable information. Given a graphical pangenome and some reasonable assumptions about linkage, paths through the graph which represent potential, real haplotypes that have not been observed can easily be imagined. Depending on the combination of features (such as single nucleotides, protein domains, genes, etc.) and the organism in question, potential haplotypes (especially using annotations and phenotypes associated with each region) could be a rich space to search for optimizations in engineered species, potential virulence in pathogens, or even to study trends in linkage.

2.5 Resolution

The fundamental unit of genomics is the base pair. While many other types of variation exist, including those described below, they can all be seen as an abstracted annotation of some sequence, or set of sequences which are comprised of bases. SVs are stretches of, usually 50 or more, bases that are the result of some mutation event and thus can be viewed as single units of variation, just as syntenic regions can be seen as individual units that comprise a pangenome. Even though they are all different resolutions of the same underlying information, they require different implementations and each defines a separate context with which to view the biological implications of the graphs and methods used.

2.5.1 Base Pair

Many studies comparing multiple genomes focus on single nucleotide variants (SNVs) or small indels, rather than larger variants like SVs or repeats. Subsequently, many approaches to graphical pangenomics focus on analyses at individual base pair resolution[1,21,24,28,29,36,45,56]. Some of these tools build a de Bruijn based pangenome graph out of *k*-mers while others build a variation graph out of a reference and VCF file of known variants, or an alignment of multiple genomes. Especially using bidirected variation graphs, operating at base pair resolution allows for any type of variant to be represented in the graph structure, including nested variation and SVs. For the most detailed alignment and variant calling, this resolution is required.

In addition to being the most detailed, graphical SNV analysis also has the highest computational costs, both in memory and in runtime. As mentioned above, the number of potential paths through a pangenome graph, each of which represents a possible haplotype, increases exponentially with the number of variants included. The ability to cover more known variants, and include those which are newly discovered, has been shown to boost performance when processing reads which contain variants[21,29,56]. Despite methods being able to prune graphs for indexing based on biologically relevant paths[54,65], there is currently a trade-off between resource requirements[29,56] and generality[21]. Performance also varies widely based on the mapping criteria which itself is potentially influenced by the read length[19]. Several studies have shown that an

increase in the variants included in pangenome graphs can lead to a decrease in the performance of alignment, potentially due to the aforementioned complexity, particularly for reads which do not contain variants[22,54].

2.5.2 Structural Variants

While reads with SNPs can often be mapped and aligned to the reference genome, reads overlapping SVs, typically defined as>50 base pairs, can be difficult to map to a linear reference since by definition they contain significant differences in a concentrated interval[17,23]. This reference bias is exacerbated by the fact that some SVs are often longer than short reads. These limitations have caused SVs to be more difficult to study and catalogue than SNPs. However, recent efforts have sought remedy this issue by discovering SVs using long read data in assemblies[21] and categorizing and storing known SVs in sequence repositories[30].

Short reads that overlap SVs can rarely be mapped to the linear reference genome with standard tools, SVs represent an area in which graphical pangenomes can help eliminate reference bias and improve detection[19]. Several tools have been built for genotyping SVs by incorporating known variation into pangenome graphs and have been shown to outperform traditional, reference-based methods[7,17,23,34,66]. The main drawback in SV level analysis is that base pair resolution is often missed, however, two of these tools[17,23] can be used to analyze both small variants and SVs. At the moment, graphical pangenome based methods and the composite models are limited by which SVs have been characterized and catalogued[57]. Further developments in the analysis of graphical pangenomes, including taking advantage or paired end reads, may allow discovery of SVs as readily as small variants are discovered in current pipelines.

2.5.3 Synteny

Synteny level analysis involves the comparison of conserved order of annotated intervals between two or more genomes. The unit of synteny can be genes, protein domains, locally collinear blocks, or any type of sequence segment which can be consistently annotated. At the SV level of resolution we discussed several methods leveraging strict mapping criteria to improve performance[35,58]. Synteny based methods may offer a vehicle to push this paradigm even further. Tools like Panaconda[74] and Ptolemy[59] apply similar logic and algorithms found in genome graph creation to output synteny graphs. Though less common, these methods replace the DNA alphabet with one based on consistent annotations which also refines the mapping criteria and speeds up calculation. This also raises the floor of the lowest modeled divergence from an SV of a given size to that of one or more annotated genome features. Due to their reliance on annotation, without further adaption these methods would not be suited to precisely the same iterative discovery and model refinement from sequencing runs. That does not mean, however, they cannot offer insightful analysis, e.g. Panaconda finds and labels inversions and translocations in bacteria that manifest at the annotation level and Kolmogorov et al.[53] recently applied a similar approach to the analysis of assembly graphs.

3 Algorithms and Software

In any modeling framework it is important to consider what the relevant assumptions are. Relative to a linear reference, genome graph models and graphical pangenomics take a broader, more general view. For human analysis, this allows us to step away from the surprising lack of diversity captured by the current human reference[18]. A broader and more flexible comparative framework may benefit many research fields where comparative genomics is currently applied including but not limited to, cancer and disease biology, synthetic biology, forensic biology, and the benefits from which new insights in those fields may derive. Because these pangenomic and graphical genome models are being created for the first time, comparative genomics is now meeting network science in a meaningful way. This means the potential for cross application is quite high but the number of discoveries enabled by this modeling change is still low. Questions have been raised many times concerning the benefits and applications of bioinformatics and computational biology. While the change those disciplines may precipitate may have been uncertain at times, they quickly became the field of modern biology itself. The authors feel this is likely to be the case for genomics and graphical pangenomes.

3.1 Mapping, Genotyping, Sequence Annotation, and Variant Calling

Four of the most common tasks in genomic and pangenomic analysis are mapping reads to a linear or graphical reference, determining the genotype of a sequenced sample, annotating sequences with relevant features, and identifying

what variants are present in a sample, whether they be SNVs, short indels, SVs, or some other type. These processes represent some of the most significant applications for graphical pangenomics, since, as described above, the bias introduced with a linear reference affects the traditional approach to all of them. Graphical pangenomics can help reduce this bias by allowing a more comprehensive and compact representation of the diversity in a population. In addition, the network inherent in these graphical models can provide a scaffold at the sequence or synteny level to help investigate and extend existing functional and process oriented annotations.

3.1.1 Small Variant Tools

Traditional, linear-reference-based aligners like BWA-MEM[32] are able to effectively map and align most reads containing SNVs, since even short read sequencing covers the relevant region. For these types of variation, pangenome graph based analysis does not offer much benefit, and may actually hurt performance due to the typical increase in time and memory needed for these methods, and the likelihood for reads to have alternative mapping positions[22,54]. However, as their size increases, small insertions and deletions (indels) are more challenging for traditional tools. Since there are many cataloged indels, for example those in dbSNP[61] for human sequences, graphical pangenomes can be constructed to represent these small variants and help reduce reference bias in mapping and alignment for reads which contain them.

Several methods for using graphical pangenomes to analyze small variants have been proposed. Some methods use colored de Bruijn graphs, which encode the origin(s) of each k–mer with color labels to preserve haplotype information. The tool Cortex uses these graphs to assemble and call variants based on coverage in regions containing bubbles[27], and is improved upon by Bubbleparse[31]. Another such tool is the de Bruijn Graph-based Aligner (deBGA), which has been shown to be faster and more accurate than linear reference based methods[36].

More recent studies have proposed variation graph based approaches for tackling this problem, including GraphTyper[16], vg[21], the Seven Bridges Graph Genome Pipeline (GGP)[56], HISAT2[29], and GraphAligner[58]. GraphTyper first maps reads to a linear reference to make an initial guess at which variation subgraphs are relevant, and then maps and aligns to those subgraphs. In this way, GraphTyper still suffers from some reference bias. All of the other tools build a pangenome graph, usually from a reference genome and VCF of known variants (vg and GraphAligner can also use arbitrary graphs, such as cactus graphs[50]). HISAT2 uses a highly efficient FM-index[20] extension to graphs (GFM)[29] which allows very efficient queries and storage compared to other tools, like vg[29]. GGP also saves time and space[22,56] in indexing by using a simpler indexing strategy than the GCSA2[64] index used by vg.

Few studies have compared these tools directly. One study reported that HISAT2 and vg had about equal sensitivity[29], while another reported that vg was superior to HISAT2 and GGP for short read mapping, and that both tools were only better than BWA-MEM on reads which contained variants[22]. While it uses the most versatile index, vg has been shown to be slower and consume more memory than other tools, especially for long reads[29,58]. GraphAligner finds its niche in being able to handle these long reads, using a simpler minimizer based seeding strategy, which limits its versatility but makes it more time and memory efficient than vg for long read analysis, with about the same performance given the constrained input[58].

3.1.2 Structural Variant Tools

The limits of variant detection due to reference bias has inspired many tools to take advantage of the inherent variation captured in graphical pangenomic references for genotyping SVs. BayesTyper[63] uses a probabilistic model to compare k–mer distributions between reads and paths in the graph[19]. Other methods include Paragraph[7], GraphTyper2[17], and vg[21,23]. GraphTyper2 and Paragraph both first align to a linear reference to determine which reads are relevant to SV containing subgraphs, and thus both suffer slightly from some reference bias. However, all have been shown to outperform traditional linear reference based methods, with vg and Paragraph seeming to perform best[7,23]. In addition, vg was shown to have higher performance when built on multiple diverse, aligned yeast genomes rather than a linear reference and VCF containing the SVs[23]. Another recent tool which can genotype SVs, Giraffe[66] (part of the vg toolkit), uses haplotype information to map and phase SVs faster than previous versions of vg, and with about the same accuracy.

For tools like vg and GraphAligner, new advancements in multiple whole genome

Phasing: The resolution of an individuals paternal and maternal chromosomes.

alignment may allow for more diverse and complete representations and analyses of structural variation. Progressive Cactus[1] is a progressive extension of the original Cactus algorithm[50] which takes as input approximate guide tree and input genomes, and was able to align over 600 mammalian genomes over two months[1]. SibeliaZ[45] builds on TwoPaCo's[46] efficient construction of compacted de Bruijn graphs to construct local collinear blocks between input genomes and then multiple sequence alignment between these blocks. SibeliaZ was shown to be faster and more memory efficient than Progressive Cactus, but has lower performance for divergent genomes[45].

3.2 Building de Bruijn Graphs

Compacted de Bruijn graphs, which merge all non-branching nodes, are a very common data structure in genomic applications for the efficient storage of contiguous segments across one or many genomes, including assembly and representing pangenomes. Colored compacted de Bruijn graphs (ccDBGs) extend the original by adding in color labels which define the sample(s) from which each k–mer originated. The efficient construction of these graphs has been a source of development over the past decade, in part to allow the analysis of many large genomes. Several tools have been developed for this purpose, many of which are able to directly build the compacted graph without needing to first build the uncompacted version. These tools include SplitMEM[42] and TwoPaCo[46]. SplitMEM uses suffix trees and suffix skips, and their topological relationship to the compacted de Bruijn graph, and was improved upon by Baier et al.[3], using the Burrows-Wheeler transform[5]. TwoPaCo uses probabilistic (Bloom filter based) and highly parallelizable methods to provide further improvements building on these previous achievements.

Bifrost[24] and Cuttlefish[28] are two, more recent, methods which provide efficient methods for building ccDBGs. Bifrost also uses a Bloom filter based approach, and was shown to be faster and use less memory than BCALM2[8], and allow faster (but more memory intensive) querying of the graph compared to Blight[41]. Bifrost is able to process whole genomes and short reads. Cuttlefish models vertices as finite-state automata and has been shown to be more time and memory efficient than Bifrost and TwoPaCo[28], but is currently only able to process whole genomes.

These graphs are only as useful as the types of analyses they enable. Blastfrost[39] is a tool built on top of Bifrost graphs for efficiently querying reads against 100,000s of bacterial genomes. It can build out subgraphs from the ccDBG based on similarity to the input sequences as an alternative to alignment based tools. This method has limitations for smaller reads and reads with increasing diversity. Two other methods, PathRacer[62] applies profile HMMs to assembly (de Bruijn graphs) to overcome the limitation of relevant segments separated over multiple contigs. BiosyntheticSPAdes[43] uses the same approach to mine biosynthetic gene clusters in bacterial assembly graphs.

4 Ecosystem of Development

As genome graphs become more accessible and well supported with respect to existing bioinformatics pipelines it is an open question as to how these constructs will be integrated into everyday biological analysis. By maintaining a genomic model that is capable of being more readily updated, genome informatics will be open to new questions regarding common practice and standards associated with this information. It has long been understood that assembled genomes represent a predictive model relative to the true DNA molecule being sequenced. Genome graphs better provide the ability to document uncertainty, heterogeneity, both clonal and pangenomic. Particularly now that long read sequencing is providing better prediction of SVs[2,6,11,25,68], single species heterogeneity can be better represented. This in turn translates to increased heterogeneity at the pangenome level.

It is possible that large sequence repositories such as NCBI[55] will adopt graphical genome structures as part of their infrastructure. This would open potential for dialogue between the evolution of graphical genome analysis and the existing reference genome ecosystem. One potential vehicle for that dialog is the recovery of reference strains from graphical genome constructs. This would simultaneously enable much more varied analysis, which could be viewed as a negative when doing literature review, while controlling for its specification. For example a hash identifier, similar to that used in revision control, could be used to designate the set of paths to take through a graph genome to construct and recover a single linear reference. For this to be possible an address system that is invariant through graphical genome updates would be necessary. There is no question that removing reference bias and

enabling topological analysis of the genome landscape will open up new insight. Having a control system in place to enable the specification of combinations of variants will be essential. Methods for annotation and phenotype association with haplotypes in this context will be subject to developing formats and their ability to model such associations.

Sequence graphs generated from most tools, such as de Bruijn graphs and other assembly graphs, are often represented using the Graphical Fragment Assembly (GFA) format. A recent paper, which also presented minigraph, proposed a new method which extends GFA to reference pangenome graphs and includes tags to defining the origin of a segment from linear genomes in a pangenome graph[33]. This format aims to provide a stable coordinate system for pangenome graphs and is used by minigraph. The vg toolkit uses a different model for pangenome graphs (called xg)[21] which allows segments to appear on multiple paths or in cycles[37], e.g. documenting identical segments being collapsed together, which is not true in the rGFA format where each segment can only be associated with one origin. A format which allows relativism's to canonical reference genomes, invariant recovery of updated reference versions, and expressivity to new structures seems desirable.

The Graphical mApping Format (GAF)[34] is a newly proposed format which extends the Pairwise mApping Format (PAF) for sequence to graph alignment and is built on top of the rGFA format[34]. This format is used by minigraph and GraphAligner[34,58]. The paper presenting vg[21] also proposed a protobuf alignment format Graph Alignment Map (GAM) which is analogous to BAM.

The GFF format is a common standard used to store annotations for genomic features and regions. A recent paper by some of the members of the vg team argued that defining annotations for sequence segments, as is done for current linear genome analysis, does not generalize well to graphs since common variants located in a given segment will not be captured by such an annotation. Therefore, they proposed a new file format, gGFF, which extends the GFF format to pangenome graphs using the notion that connected subgraphs should be used as the annotated unit[37].

5 Conclusion

As the field of graphical pangenomes continues to mature, more tools and innovations will bring this class of genomic analysis closer to the computational requirements of current linear tools like BWA-MEM[32], while decreasing reference bias and allowing the efficient incorporation of a growing number of known variants. Though this field is rapidly evolving, several limitations still exist, such as incorporating haplotype information to common analysis methods, defining a coordinate system which is amenable to arbitrary and nested variation types and incremental updates and revisions, shared and standard formats, ease of use and accessibility. We hope that this review of current methods and updates will serve as a useful checkpoint and reference for those working in and interested in the field of graphical pangenomes.

Publisher's Note

Declarations

Conflict of interest

A. Warren is the primary author and developer of the Panaconda software.

Received: 16 June 2021 Accepted: 7 July 2021
Published online: 24 August 2021

References

1. Armstrong J, Hickey G, Diekhans M, Fiddes IT, Novak AM, Deran A, Fang Q, Xie D, Feng S, Stiller J, Genereux D, Johnson J, Marinescu VD, Alföldi J, Harris RS, Lindblad-Toh K, Haussler D, Karlsson E, Jarvis ED, Zhang G, Paten B (2020) Progressive Cactus is a multiple-genome aligner for the thousand-genome era. Nature 587(7833):246–251. https://doi.org/10.1038/s41586-020-2871-y
2. Audano PA, Sulovari A, Graves-Lindsay TA, Cantsilieris S, Sorensen M, Welch AE, Dougherty ML, Nelson BJ, Shah A, Dutcher SK, Warren WC, Magrini V, McGrath SD, Li YI, Wilson RK, Eichler EE (2019) Characterizing the major structural variant alleles of the human genome. Cell 176(3):663–675.e19. https://doi.org/10.1016/j.cell.2018.12.019
3. Baier U, Beller T, Ohlebusch E (2016) Graphical pangenome analysis with compressed suffix trees and the Burrows–Wheeler transform. Bioinformatics 32(4):497–504. https://doi.org/10.1093/bioinformatics/btv603
4. Barra V, Fachinetti D (2018) The dark side of centromeres: types, causes and consequences of structural abnormalities implicating centromeric DNA. Nat Commun. https://doi.org/10.1038/s41467-018-06545-y

5. Burrows M, Wheeler D (1994) A block-sorting lossless data compression algorithm. DigiItal SRC Research Report
6. Chaisson MJ, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F, Antonacci F, Surti U, Sandstrom R, Boitano M, Landolin JM, Stamatoyannopoulos JA, Hunkapiller MW, Korlach J, Eichler EE (2015) Resolving the complexity of the human genome using single-molecule sequencing. Nature 517(7536):608–611. https://doi.org/10.1038/nature13907
7. Chen S, Krusche P, Dolzhenko E, Sherman RM, Petrovski R, Schlesinger F, Kirsche M, Bentley DR, Schatz MC, Sedlazeck FJ, Eberle MA (2019) Paragraph: A graph-based structural variant genotyper for short-read sequence data. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://doi.org/10.1101/635011, https://www.biorxiv.org/content/10.1101/635011v1
8. Chikhi R, Limasset A, Medvedev P (2016) Compacting de Bruijn graphs from sequencing data quickly and in low memory. Bioinform (Oxf Engl) 32(12):i201–i208. https://doi.org/10.1093/bioinformatics/btw279
9. Colquhoun RM, Hall MB, Lima L, Roberts LW, Malone KM, Hunt M, Letcher B, Hawkey J, George S, Pankhurst L, Iqbal Z (2020) Nucleotide-resolution bacterial pangenomics with reference graphs. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/2020.11.12.380378v3
10. Consortium CPG (2016) Computational pan-genomics: status, promises and challenges. Brief Bioinform 19(1):118–135. https://doi.org/10.1093/bib/bbw089
11. Cretu Stancu M, van Roosmalen MJ, Renkens I, Nieboer MM, Middelkamp S, de Ligt J, Pregno G, Giachino D, Mandrile G, Espejo Valle-Inclan J, Korzelius J, de Bruijn E, Cuppen E, Talkowski ME, Marschall T, de Ridder J, Kloosterman WP (2017) Mapping and phasing of structural variation in patient genomes using nanopore sequencing. Nat Commun 8(1):1326. https://doi.org/10.1038/s41467-017-01343-4
12. Dilthey A, Cox C, Iqbal Z, Nelson MR, McVean G (2015) Improved genome inference in the MHC using a population reference graph. Nat Genet 47(6):682–688. https://doi.org/10.1038/ng.3257
13. Dolzhenko E, Deshpande V, Schlesinger F, Krusche P, Petrovski R, Chen S, Emig-Agius D, Gross A, Narzisi G, Bowman B, Scheffler K, van Vugt JJFA, French C, Sanchis-Juan A, Ibáñez K, Tucci A, Lajoie BR, Veldink JH, Raymond FL, Taft RJ, Bentley DR, Eberle MA (2019) ExpansionHunter: a sequence-graph-based tool to analyze variation in short tandem repeat regions. Bioinformatics 35(22):4754–4756. https://doi.org/10.1093/bioinformatics/btz431
14. Durbin R (2014) Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT). Bioinformatics 30(9):1266–1272. https://doi.org/10.1093/bioinformatics/btu014
15. Earl D, Nguyen N, Hickey G, Harris RS, Fitzgerald S, Beal K, Seledtsov I, Molodtsov V, Raney BJ, Clawson H, Kim J, Kemena C, Chang JM, Erb I, Poliakov A, Hou M, Herrero J, Kent WJ, Solovyev V, Darling AE, Ma J, Notredame C, Brudno M, Dubchak I, Haussler D, Paten B (2014) Alignathon: a competitive assessment of whole-genome alignment methods. Genome Res 24(12):2077–2089. https://doi.org/10.1101/gr.174920.114
16. Eggertsson HP, Jonsson H, Kristmundsdottir S, Hjartarson E, Kehr B, Masson G, Zink F, Hjorleifsson KE, Jonasdottir A, Jonasdottir A, Jonsdottir I, Gudbjartsson DF, Melsted P, Stefansson K, Halldorsson BV (2017) Graphtyper enables population-scale genotyping using pangenome graphs. Nat Genet 49(11):1654–1660. https://doi.org/10.1038/ng.3964
17. Eggertsson HP, Kristmundsdottir S, Beyter D, Jonsson H, Skuladottir A, Hardarson MT, Gudbjartsson DF, Stefansson K, Halldorsson BV, Melsted P (2019) GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. Nat Commun 10(1):5402. https://doi.org/10.1038/s41467-019-13341-9
18. Eisfeldt J, Mårtensson G, Ameur A, Nilsson D, Lindstrand A (2020) Discovery of Novel Sequences in 1,000 Swedish Genomes. Mol Biol Evolut 37(1):18–30. https://doi.org/10.1093/molbev/msz176
19. Eizenga J, Novak A, Sibbesen J, Heumos S, Ghaffaari A, Hickey G, Chang X, Seaman J, Rounthwaite R, Ebler J, Rautiainen M, Garg S, Paten B, Marschall T, Sirén J, Garrison E (2020) Pangenome graphs. Ann Rev Genom Hum Genet 21:139–162. https://doi.org/10.1146/annurev-genom-120219-080406
20. Ferragina P, Manzini G (2000) Opportunistic data structures with applications. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 390–398. IEEE Comput. Soc, Redondo Beach, CA, USA. https://doi.org/10.1109/SFCS.2000.892127, http://ieeexplore.ieee.org/document/892127/
21. Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, Dawson ET, Jones W, Garg S, Markello C, Lin MF, Paten B, Durbin R (2018) Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nat Biotechnol 36(9):875–879. https://doi.org/10.1038/nbt.4227
22. Grytten I, Rand KD, Nederbragt AJ, Sandve GK (2020) Assessing graph-based read mappers against a baseline approach highlights strengths and weaknesses of current methods. BMC Genom 21(1):282. https://doi.org/10.1186/s12864-020-6685-y
23. Hickey G, Heller D, Monlong J, Sirén J, Dawson ET, Garrison E, Novak AM, Paten B (2020) Genotyping structural variants in pangenome graphs using the vg toolkit. Genome Biol 21(1):35. https://doi.org/10.1186/s13059-020-1941-7
24. Holley G, Melsted P (2020) Bifrost: highly parallel construction and indexing of colored and compacted de

Bruijn graphs. Genome Biol 21(1):249. https://doi.org/10.1186/s13059-020-02135-8

25. Huddleston J, Chaisson MJ, Steinberg KM, Warren W, Hoekzema K, Gordon D, Graves-Lindsay TA, Munson KM, Kronenberg ZN, Vives L, Peluso P, Boitano M, Chin CS, Korlach J, Wilson RK, Eichler EE (2017) Discovery and genotyping of structural variation from long-read haploid genome sequence data. Genome Res 27(5):677–685. https://doi.org/10.1101/gr.214007.116
26. Iantorno S, Gori K, Goldman N, Gil M, Dessimoz C (2014) Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. Methods Mol Biol 1079:59–73. https://doi.org/10.1007/978-1-62703-646-7_4
27. 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(2):226–232. https://doi.org/10.1038/ng.1028
28. Khan J, Patro R (2020) Cuttlefish: fast, parallel, and low-memory compaction of de bruijn graphs from large-scale genome collections. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/2020.10.21.349605v1
29. 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(8):907–915. https://doi.org/10.1038/s41587-019-0201-4
30. Lappalainen I, Lopez J, Skipper L, Hefferon T, Spalding JD, Garner J, Chen C, Maguire M, Corbett M, Zhou G, Paschall J, Ananiev V, Flicek P, Church DM (2013) dbVar and DGVa: public archives for genomic structural variation. Nucl Acids Res 41(Database issue): D936–D941. https://doi.org/10.1093/nar/gks1213. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531204/
31. Leggett RM, Ramirez-Gonzalez RH, Verweij W, Kawashima CG, Iqbal Z, Jones JDG. Caccamo M, MacLean D (2013) Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de bruijn graphs. PLoS One. https://doi.org/10.1371/journal.pone.0060058, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3607606/
32. Li H (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio]
33. Li H (2016) Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics 32(14):2103–2110. https://doi.org/10.1093/bioinformatics/btw152
34. Li H, Feng X, Chu C (2020) The design and construction of reference pangenome graphs. arXiv:2003.06079 [q-bio]
35. Li H, Feng X, Chu C (2020) The design and construction of reference pangenome graphs with minigraph. Genome Biol 21(1):265. https://doi.org/10.1186/s13059-020-02168-z
36. Liu B, Guo H, Brudno M, Wang Y (2016) deBGA: read alignment with de Bruijn graph-based seed and extension. Bioinformatics 32(21):3224–3232. https://doi.org/10.1093/bioinformatics/btw371
37. Llamas B, Narzisi G, Schneider V, Audano PA, Biederstedt E, Blauvelt L, Bradbury P, Chang X, Chin CS, Fungtammasan A, Clarke WE, Cleary A, Ebler J, Eizenga J, Sibbesen JA, Markello CJ, Garrison E, Garg S, Hickey G, Lazo GR, Lin MF, Mahmoud M, Marschall T, Minkin I, Monlong J, Musunuri RL, Sagayaradj S, Novak AM, Rautiainen M, Regier A, Sedlazeck FJ, Siren J, Souilmi Y, Wagner J, Wrightsman T, Yokoyama TT, Zeng Q, Zook JM, Paten B, Busby B (2019) A strategy for building and using a human reference pangenome. F1000 Res 8:1751. https://doi.org/10.12688/f1000research.19630.1https://f1000research.com/articles/8-1751/v1
38. Lu TY, Consortium THGSV, Chaisson M (2020) Profiling variable-number tandem repeat variation across populations using repeat-pangenome graphs. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/2020.08.13.249839v1
39. Luhmann N, Holley G, Achtman M (2020) BlastFrost: Fast querying of 100,000s of bacterial genomes in Bifrost graphs. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/2020.01.21.914168v1
40. Maciuca S, del OjoElias C, McVean G, Iqbal Z (2016) A natural encoding of genetic variation in a Burrows–Wheeler transform to enable mapping and genome inference. In: Frith M, Storm Pedersen CN (eds) Algorithms in Bioinformatics, Lecture Notes in Computer Science, pp. 222–233. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-43681-4_18
41. Marchet C, Kerbiriou M, Limasset A (2020) Efficient exact associative structure for sequencing data. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/546309v3
42. Marcus S, Lee H, Schatz MC (2014) SplitMEM: a graphical algorithm for pan-genome analysis with suffix skips. Bioinformatics 30(24):3476–3483. https://doi.org/10.1093/bioinformatics/btu756
43. Meleshko D, Mohimani H, Tracanna V, Hajirasouliha I, Medema MH, Korobeynikov A, Pevzner PA (2019) BiosyntheticSPAdes: reconstructing biosynthetic gene clusters from assembly graphs. Genome Res 29(8):1352–1362. https://doi.org/10.1101/gr.243477.118
44. Miga KH (2019) Centromeric satellite DNAs: hidden sequence variation in the human population. Genes 10:352
45. Minkin I, Medvedev P (2019) Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. bioRxiv. (**Publisher: Cold Spring**

Harbor Laboratory Section: New Results). https://www.biorxiv.org/content/10.1101/548123v1

46. Minkin I, Pham S, Medvedev P (2017) TwoPaCo: an efficient algorithm to build the compacted de Bruijn graph from many complete genomes. Bioinform (Oxf Engl) 33(24):4024–4032. https://doi.org/10.1093/bioinformatics/btw609
47. Notredame C (2007) Recent evolutions of multiple sequence alignment algorithms. PLOS Comput Biol 3(8):e123. https://doi.org/10.1371/journal.pcbi.0030123
48. Novak AM, Garrison E, Paten B (2017) A graph extension of the positional Burrows-Wheeler transform and its applications. Algorithms Mole Biolo: AMB. https://doi.org/10.1186/s13015-017-0109-9
49. Novak AM, Hickey G, Garrison E, Blum S, Connelly A, Dilthey A, Eizenga J, Elmohamed MAS, Guthrie S, Kahles A, Keenan S, Kelleher J, Kural D, Li H, Lin MF, Miga K, Ouyang N, Rakocevic G, Smuga-Otto M, Zaranek AW, Durbin R, McVean G, Haussler D, Paten B (2017) Genome Graphs. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/101378v1
50. Paten B, Diekhans M, Earl D, John JS, Ma J, Suh B, Haussler D (2011) Cactus graphs for genome comparisons. J Comput Biol 18(3):469–481. https://doi.org/10.1089/cmb.2010.0252
51. Paten B, Eizenga JM, Rosen YM, Novak AM, Garrison E, Hickey G (2018) Superbubbles, Ultrabubbles, and Cacti. J Comput Biol 25(7):649–663. https://doi.org/10.1089/cmb.2017.0251
52. Paten B, Novak AM, Eizenga JM, Garrison E (2017) Genome graphs and the evolution of genome inference. Genome Res 27(5):665–676. https://doi.org/10.1101/gr.214155.116
53. Polevikov E, Kolmogorov M (2019) Synteny paths for assembly graphs comparison. In: Huber KT, Gusfield D (eds) 19th International Workshop on Algorithms in Bioinformatics (WABI), Leibniz International Proceedings in Informatics (LIPIcs), vol. 143, pp. 24:1–24:14. Schloss Dagstuhl–Leibniz–Zentrum fuer Informatik, Dagstuhl, Germany, ISSN: 1868-8969. https://doi.org/10.4230/LIPIcs.WABI.2019.24, http://drops.dagstuhl.de/opus/volltexte/2019/11054
54. Pritt J, Chen NC, Langmead B (2018) FORGe: prioritizing variants for graph genomes. Genome Biol 19(1):220. https://doi.org/10.1186/s13059-018-1595-x
55. Pruitt KD, Tatusova T, Maglott DR (2005) NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res 33(suppl–1):D501–D504. https://doi.org/10.1093/nar/gki025
56. Rakocevic G, Semenyuk V, Lee WP, Spencer J, Browning J, Johnson IJ, Arsenijevic V, Nadj J, Ghose K, Suciu MC, Ji SG, Demir G, Li L, Toptas BC, Dolgoborodov A, Pollex B, Spulber I, Glotova I, Kómár P, Stachyra AL, Li Y, Popovic M, Källberg M, Jain A, Kural D (2019) Fast and accurate genomic analyses using genome graphs. Nat Genet 51(2):354–362. https://doi.org/10.1038/s41588-018-0316-4
57. Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics 28(18):i333–i339. https://doi.org/10.1093/bioinformatics/bts378
58. Rautiainen M, Marschall T (2020) GraphAligner: rapid and versatile sequence-to-graph alignment. Genome Biol 21(1):253. https://doi.org/10.1186/s13059-020-02157-2
59. Salazar AN, Abeel T (2018) Approximate, simultaneous comparison of microbial genome architectures via syntenic anchoring of quiver representations. Bioinformatics 34(17):i732–i742. https://doi.org/10.1093/bioinformatics/bty614
60. Sherman RM, Salzberg SL (2020) Pan-genomics in the human genome era. Nat Rev Genet 21(4):243–254. https://doi.org/10.1038/s41576-020-0210-7
61. Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K (2001) dbSNP: the NCBI database of genetic variation. Nucl Acids Res 29(1):308–311. https://doi.org/10.1093/nar/29.1.308
62. Shlemov A, Korobeynikov A (2019) PathRacer: racing profile HMM paths on assembly graph. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/562579v1
63. Sibbesen JA, Maretty L, Krogh A (2018) Accurate genotyping across variant classes and lengths using variant graphs. Nat Genet 50(7):1054–1059. https://doi.org/10.1038/s41588-018-0145-5
64. Sirén J (2017) Indexing variation graphs. In: 2017 Proceedings of the Ninteenth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 13–27. https://doi.org/10.1137/1.9781611974768.2, http://arxiv.org/abs/1604.06605
65. Sién J, Garrison E, Novak AM, Paten B, Durbin R (2018) Haplotype-aware graph indexes. arXiv:1805.03834 [cs]
66. Sirén J, Monlong J, Chang X, Novak AM, Eizenga JM, Markello C, Sibbesen J, Hickey G, Chang PC, Carroll A, Haussler D, Garrison E, Paten B (2020) Genotyping common, large structural variations in 5,202 genomes using pangenomes, the Giraffe mapper, and the vg toolkit. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://www.biorxiv.org/content/10.1101/2020.12.04.412486v1
67. Slotkin RK (2018) The case for not masking away repetitive DNA. Mob DNA 9(1):15. https://doi.org/10.1186/s13100-018-0120-9
68. Spies N, Weng Z, Bishara A, McDaniel J, Catoe D, Zook JM, Salit M, West RB, Batzoglou S, Sidow A (2017) Genome-wide reconstruction of complex structural

variants using read clouds. Nat Methods 14(9):915–920. https://doi.org/10.1038/nmeth.4366

69. Tettelin H, Masignani V, Cieslewicz MJ, Donati C, Medini D, Ward NL, Angiuoli SV, Crabtree J, Jones AL, Durkin AS, Deboy RT, Davidsen TM, Mora M, Scarselli M, Margarity Ros I, Peterson JD, Hauser CR, Sundaram JP, Nelson WC, Madupu R, Brinkac LM, Dodson RJ, Rosovitz MJ, Sullivan SA, Daugherty SC, Haft DH, Selengut J, Gwinn ML, Zhou L, Zafar N, Khouri H, Radune D, Dimitrov G, Watkins K, O'Connor KJB, Smith S, Utterback TR, White O, Rubens CE, Grandi G, Madoff LC, Kasper DL, Telford JL, Wessels MR, Rappuoli R, Fraser CM (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". Proc Natl Acad Sci USA 102(39):13950–13955. https://doi.org/10.1073/pnas.0506758102
70. Valenzuela D, Norri T, Välimäki N, Pitkänen E, Mäkinen V (2018) Towards pan-genome read alignment to improve variation calling. BMC Genom 19(Suppl 2):87. https://doi.org/10.1186/s12864-018-4465-8
71. Vernikos G, Medini D, Riley DR, Tettelin H (2015) Ten years of pan-genome analyses. Curr Opin Microbiol 23:148–154. https://doi.org/10.1016/j.mib.2014.11.016
72. Vernikos GS (2020) A review of pangenome tools and recent studies. In: Tettelin H, Medini D (eds) The Pangenome. Springer, Cham, pp 89–112 http://link.springer.com/10.1007/978-3-030-38281-0_4
73. Vollmers J, Wiegand S, Kaster AK (2017) Comparing and evaluating metagenome assembly tools from a microbiologist's perspective - not only size matters!. PLoS One. https://doi.org/10.1371/journal.pone.0169662, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5242441/
74. Warren AS, Davis JJ, Wattam AR, Machi D, Setubal JC, Heath LS (2017) Panaconda: application of pan-synteny graph models to genome content analysis. bioRxiv. (**Publisher: Cold Spring Harbor Laboratory Section: New Results**). https://doi.org/10.1101/215988. https://www.biorxiv.org/content/10.1101/215988v1
75. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18(5):821–829. https://doi.org/10.1101/gr.074492.107

Joseph Outten is a researcher at the Biocomplexity Institute at UVA, and his interests include machine learning, algorithms, systems biology, and complexity and information theory.

Andrew Warren is a Research Assistant Professor at the Biocomplexity Institute and Initiative at the University of Virginia. His interests include computational biology, disease bioinformatics, graphical genomes, ontologies, rnaseq, metagenomics, machine learning, pangenomics, and biosurveillance.