Investigating the topological motifs of inversions in pangenome graphs

Sandra Romain[1], Siegfried Dubois[1], Fabrice Legeai[1,2], and Claire Lemaitre[1]

[1] *Univ Rennes, Inria, CNRS, IRISA - UMR 6074, F-35000 Rennes, France*
[2] *IGEPP, INRAE, Institut Agro, University of Rennes, 35653 Le Rheu, France*

Abstract

Recent technological advances have accelerated the production of high-quality genome assemblies within species, driving the growing use of pangenome graphs in genetic diversity analyses. These graphs reduce reference bias in read mapping and enhance variant discovery and genotyping from SNPs to Structural Variants (SVs). In pangenome graphs, variants appear as bubbles, which can be detected by dedicated bubble calling tools. Although these tools report essential information on the variant bubbles, such as their position and allele walks in the graph, they do not annotate the type of the detected variants. While simple SNPs, insertions, and deletions are easily distinguishable by allele size, large balanced variants like inversions are harder to differentiate among the large number of unannotated bubbles. In fact, inversions and other types of large variants remain underexplored in pangenome graph benchmarks and analyses.

In this work we focused on inversions, which have been drawing renewed attention in evolutionary genomics studies in the past years, and aimed to assess how this type of variant is handled by state of the art pangenome graphs pipelines. We identified two distinct topological motifs for inversion bubbles: one path-explicit and one alignment-rescued, and developed a tool to annotate them from bubble-caller outputs. We constructed pangenome graphs with both simulated data and real data using four state of the art pipelines, and assessed the impact of inversion size, punctual genome divergence and haplotype number on inversion representation and accuracy.

Our results reveal substantial differences between pipelines, with many inversions either misrepresented or lost. Most notably, recovery rates remain strikingly low, even with the most simple simulated genome sets, highlighting major challenges in analyzing inversions in pangenomic approaches.

1 Introduction

The recent increase in the number of high-quality assembled genomes for a same species call for a switch in how we represent the genomic reference in genetic diversity analyses, from a single linear reference genome to variation and pangenome graphs. The interest of pangenome graphs resides in their representation of genomic variation between multiple genomes, allowing a diminution of reference bias when mapping reads, thus improving variant discovery and genotyping for the whole

range of variant sizes and types (Garrison et al., 2018; Hickey et al., 2024). The current tools for pangenome graph construction rely on genome alignments, either by iterative pairwise alignments from the reference genome for Minigraph (Li et al., 2020) and its derived pipeline Minigraph-Cactus (Hickey et al., 2024), by reference-free pairwise alignments for PGGB (Garrison et al., 2023), or by tree-guided alignments for Progressive Cactus (Armstrong et al., 2020). The standard pipeline in pangenome-graph based variant studies, performed in recent studies such as on bovine (Leonard et al., 2023), wild grape (Cochetel et al., 2023) and human (Liao et al., 2023) genomes, comprises the construction of a graph from a collection of genomes, the identification of variants in the graph and their genotyping in other individuals or populations from read mapping onto the graph. The types of represented variants in the graphs differ between the tools, as Minigraph only aims to represent Structural Variants (SVs), typically variants that are larger than 50 bp, while the other tools also represent smaller variants in their graphs. Variant identification in pangenome graphs consists in analysing the graph topology to find topological motifs called *bubbles*. Bubbles are formed when (at least) two genome paths diverge in the graph due to sequence differences, and meet again further on after the differing portion. For instance, a SNP is represented in the graph by a small bubble where each allele carried in the input genomes forms a diverging path through a node of length 1 bp, while a SV is represented by a larger bubble where at least one of the diverging paths has a length of at least 50 bp. Pangenome graphs may also contain more complex bubbles such as nested bubbles, which can be the result of nested variants (*e.g.* small variants inside an insertion). The main tools for bubble detection are *vg deconstruct* (Liao et al., 2023), *BubbleGun* (Dabbaghie et al., 2022) and *gfatools* (Li, 2019), the former being the most used one as it is included in Minigraph-Cactus and PGGB pipelines. Although these tools report essential information on the variant bubbles detected in the graphs (*e.g.* position, alleles), they do not annotate the type of the detected variants. Without annotation of the bubbles' variant category, it is difficult to count the number, type and size distribution of the SVs represented in a pangenome graph. The annotation of SNP bubbles does not pose much of a problem as they have distinctive and exclusive allele path sizes of 1 bp. Likewise, bubbles having one of their allele paths with a size of 0 bp, a distinctive feature of insertions and deletions, can also be confidently annotated. However, all the remaining bubbles, in particular those with large sized paths, are more challenging to confidently annotate as a specific SV type, as their path size may correspond to more than one SV type. The large size of SVs makes them also more likely to contain nested bubbles, which is not that straightforward to represent in a VCF format. Vcfwave, from the vcflib suite (Garrison et al., 2022), can re-align the allele sequences of the large bubbles reported in the *vg deconstruct* VCF to break down all their nested bubbles into a series of insertions, deletions and SNPs. Additionally, a recent work focused on a specific bubble motif generated by inverted repeats or hairpin-like structures (Mwaniki et al., 2024).

Among Structural Variants, inversions happen to be less frequent in genomes than other types of variants such as insertions and deletions, but can reach particularly large sizes. In the human genome, inversions account for 0.4% of the diploid genome, with a median size of 17.6 kbp and a maximum size over 5 Mbp (Porubsky et al., 2022). Nevertheless, inversions have an exceptional potential to promote adaptation and speciation. When in heterozygous state, they can locally reduce the effective rate of recombination in genomes, which can lead to the spread of adaptative allele combinations in populations or to genetic isolation through mutation accumulation (Kirkpatrick, 2010; Huang and Rieseberg, 2020; Berdan et al., 2023). Large inversions in particular, potentially including more genes, may have a higher probability of capturing non neutral allele combinations (Wellenreuther and Bernatchez, 2018). However, they are known to be more challenging to charac-

terize and genotype than other types of SVs (through sequence based approaches) due to complex and often repetitive genomic context (Liu et al., 2024; Sanders et al., 2016).

While a general improvement of structural variant analysis using pangenome graphs has been reported, inversions are often overlooked in pangenome graph benchmarks and analyses. In Liao et al. (2023) and Leonard et al. (2023) benchmarks of human and bovine pangenomes respectively or Cochetel et al. (2023) analysis of the wild-grape pangenome, SVs were either evaluated as a whole or only insertions and deletions were reported. Only the Hickey et al. (2024) benchmark paper reports some inversion counts, but only for the *Drosophila* pangenome. This could very well be a reflection of the difficulty of distinguishing the different SV types among graph bubbles of large size.

Because accurate representation of all types of variants is a crucial component to make the most out of pangenome graphs, there is a need to assess how inversions are handled by current state of the art pangenome graph pipelines (construction and analysis tools).

In this paper, we explore how inversions are represented in pangenome graphs, as well as the factors impacting their representation and thus the ability to detect them in these graphs. To this end, we constructed pangenome graphs on several simulated genome sets, as well as on real human and butterfly sets of genomes, using the state of the art pipelines. We propose two different topology motifs for inversions and annotated them among the bubbles output by the pipelines. We found that each pangenome graph pipeline presents a unique pattern of inversion topologies distribution, and that inversions are not always represented explicitly in the paths of the bubbles and need local re-alignment to be identified among the graph bubbles. Additionally, we propose INVPG-annot, a new tool to annotate, among the variant bubbles identified in a pangenome graph, those that are likely to represent inversion variants.

2 Material and Methods

2.1 Expected topologies of inversions

In a pangenome graph, the nodes are labeled with genomic sequences, and the directed edges between nodes represent the adjacencies of the sequences in genomes. The nodes can be accessed in two orientations, $forward$ when accessing the node's label sequence or $reverse$ when accessing the reverse complement of the node's label sequence. The haplotypes used to construct the graph are represented by paths in the graph (succession of nodes and their traversal orientations). In such a graph, a shared sequence between several genomes at a given locus is represented as a single node that is traversed by several paths, while a genomic variant is represented by distinct nodes connected in a particular topological motif, called a *bubble*. A bubble is a connected portion of the graph (subgraph) where two nodes, referred as a source node and a sink node, are linked by at least two distinct paths. A single bubble can represent a single variant or several overlapping or nested variants, with all distinct paths between the source and sink nodes identifying all possible alleles or allele combinations of the underlying variant(s).

Contrary to other SV types (*e.g.* insertions, deletions), an inversion does not imply a modification of sequence content between alleles. As such, in the simplest event of inversion (without any inner sequence variation), we can expect both allele sequences to be represented as a single node n_I in the graph, that is traversed in opposite directions (forward and reverse) by the ancestral (p_A) and inverted (p_I) allele paths (Fig. 1a). As inversions can cover large portions of the genome, it is most likely inversion loci harbour inner smaller variants (*e.g.* SNPs, indels). If the graph building

algorithm represents small variants in the graph, those will break the inversion locus into multiple nodes and the list of nodes traversed by p_A and p_I will differ (a simple example with one SNP is illustrated in Fig. 1b). Nevertheless, we expect the cumulative length of the nested variants to be much smaller than the total length of the inversion, allowing for the existence of nodes that are common and traversed in opposite directions between p_A and p_I. In both simplistic and realistic cases, the inversion event should be explicitly represented by the paths of the bubble, and so we define this topology type as a "path-explicit" inversion representation.

However, if an inversion is not 'recognized' either during the genome alignment step or during the PG graph construction, both ancestral and inverted alleles could be represented as unrelated sequences in the graph. In the simplest cases with two haplotypes, the locus could be represented by two entirely disjoint paths, one traversing a node n_B carrying the ancestral version of the sequence, the other traversing a node n_Z carrying the inverted version of the sequence (Fig. 1 d,e). In this case, in order to identify the inversion, one path sequence must be compared in its forward orientation to the reverse complement of the other path sequence, and so we define this topology type as an "alignment-rescued" inversion representation.

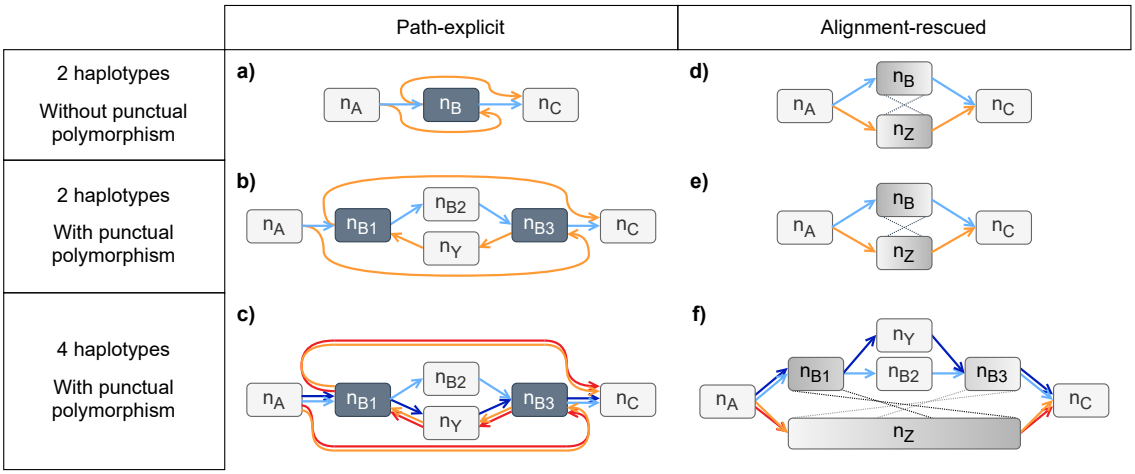

Figure 1: Illustration of the expected types of bubble topologies for inversions in pangenome graphs. Arcs of the graphs are colored according to the presence (orange and red) or absence (blue) of the inversion. The examples on the left illustrate cases where an inversion is explicitly represented in the paths of a bubble ("path-explicit"), with the dark grey colored nodes being the ones that are traversed in both directions **(a, b, c)**. The examples on the right illustrate cases where the two alleles of an inversion are represented by distinct nodes and paths ("alignment-rescued"), with grey-shaded nodes indicating pairs of nodes that are the reverse complement of each other **(d, e, f)**. For each expected topology type, different levels of complexity of the graph are presented: a first level with two haplotypes and no punctual polymorphism inside the inversion **(a, d)**, a second level with two haplotypes and a SNP between the two inversion alleles (**b, e**), and a third level with four haplotypes, one of which (*dark blue*) carries the reference allele of the inversion and the alternative allele of the SNP, meaning that the SNP alleles are not in perfect linkage disequilibrium with the inversion (**c, f**).

2.2 Inversion annotation method

We now present our method to identify, among the bubbles found in a pangenome graph, the ones that correspond to these topologies (see an overview in Fig. 2). Our method starts with bubbles that have already been found and aims at annotating them, since the main pangenome graph pipelines (*i.e.* Minigraph-Cactus and PGGB) output bubbles at the same time as they construct the graph. The graph bubbles are reported in the conventional VCF format (in a version adapted to pangenome graphs) by applying the *vg deconstruct* tool (v.1.61.0) on the Cactus, Minigraph-Cactus and PGGB graphs and the *minigraph-call pipeline* (described in Minigraph documentation with Minigraph v.0.21) on the Minigraph graphs. The pangenome VCF reports the classical information of start position of the variant on a chosen reference genome, and the reference and allele sequences. Additional information specific to the graph is also given by *vg deconstruct* and *minigraph-call pipeline*, such as the bubble's source and sink node IDs or the path of each allele of the bubble, given as an ordered and oriented list of nodes. As those VCFs readily provide an extensive description of the bubble reference and alternative paths, we opted to use them as a base on which to annotate inversions among the bubbles.

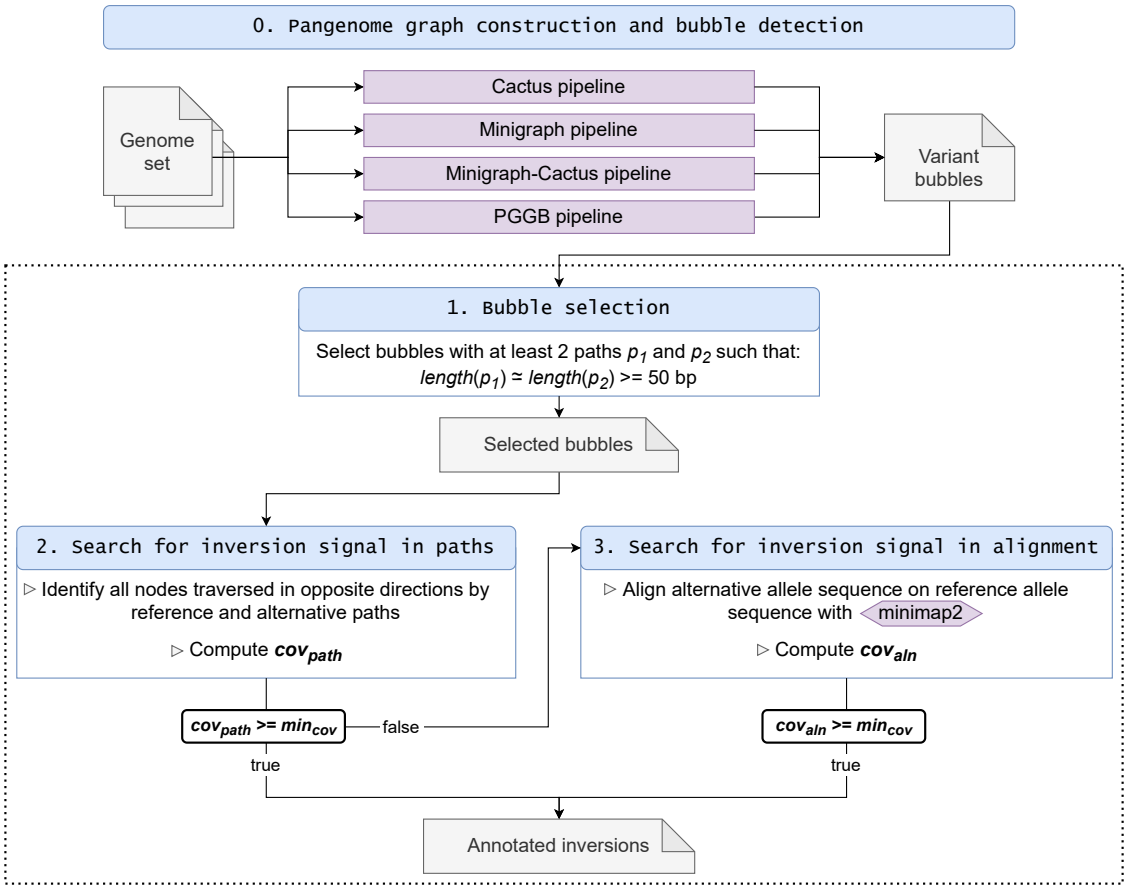

Figure 2: Illustration of the inversion annotation pipeline and method (dotted box). First, a pangenome graph is constructed from a genome set and analyzed by *vg deconstruct* (Cactus, Minigraph-Cactus and PGGB pipelines) or the *minigraph-call pipeline* (Minigraph pipeline) to obtain a list of the variant bubbles represented in the graph. Then, a first selection of bubbles is done to keep only potential inversion bubbles. These bubbles are then searched for path-based and alignment-based inversion signals.

2.2.1 Bubble selection

In order to limit the amount of bubbles to analyze, we select bubbles resembling to balanced SVs by first extracting bubbles with at least two alleles such that their length are similar and both greater than 50 bp. We consider two alleles a_0 and a_1, of length l_0 and l_1, to be of similar length if $|l_0 - l_1| \leq max(l_0, l_1) * \alpha$, with $0 \leq \alpha \ll 1$. The parameter α allows for leniency on the tolerated length difference of inversion alleles, as such difference can originate from innate biological sequence divergence but also from possible computational artefact. In the experiments, we used $\alpha = 0.1$ for all real and simulated datasets.

2.2.2 Search for inversion signal in paths and alignments

In order to annotate inversions among the selected variant bubbles, we first look for inversion signals in the paths of the bubbles. For each bubble, all alternative alleles are treated independently and compared to the reference allele.

We consider a path-based inversion signal as a common node between the alternative and reference allele paths that is traversed in opposite ways by the two paths. In order to identify such nodes, we convert the reference and alternative paths into two ordered lists $ord_{ref} = [x_1, .., x_n]$ and $ord_{alt} = [y_1, .., y_m]$ of integer node identifiers, which are positive if the node is traversed in forward and negative if the node is traversed in reverse by the path. Then for each node identifier x_i in ord_{ref}, we save x_i as a path-based inversion signal if $-x_i$ is found in ord_{alt}. For each alternative allele, we calculate a path-based signal score cov_{path}, which is the fraction of the reference allele's length covered by path-based inversion signals, with the following formula:

$$cov_{path} = \frac{\sum_{i=1}^{n} length(x_i) \times \mathbf{1}_{\exists j \text{ such that } y_j = -x_i}}{\sum_{i=1}^{n} length(x_i)}$$

If cov_{path} is greater than a user-defined threshold (min_{cov}, with default value of 0.5) for at least one alternative path of the bubble, the bubble is annotated as a path-explicit inversion. If not, we then look for inversion signals in the alignment of the bubble alternative allele sequences to the reference allele. We consider an alignment-based inversion signal as a reverse alignment between the alternative and reference allele sequences. We align all alternative alleles sequences on the reference allele sequence using minimap2 v2.15 with the parameters '-cx asm20 –cs -r2k'. For each alternative allele, we identify all its reverse alignments on the reference allele and calculate an alignment-based signal score cov_{aln} as the length fraction of the reference allele covered by reverse alignments. Considering $A = (a_1, ..a_k)$ the set of reverse alignments, this gives the following formula:

$$cov_{aln} = \frac{\sum_{a_1}^{a_k} length(a_i)}{\sum_{i=1}^{n} length(x_i)}$$

If cov_{aln} is greater than the min_{cov} threshold for at least one of the alternative paths, the bubble is annotated as an alignment-rescued inversion.

2.2.3 Implementation

The presented method of inversion annotation of pangenome graph bubbles is implemented in Python and is available on github (`https://github.com/SandraLouise/INVPG_annot`). It takes as input a GFA file of the pangenome graph as well as the corresponding VCF containing the bubbles (*vg deconstruct* or *mingraph-call* formats, reporting allele walks through the bubbles), and outputs a BED file containing the set of inversion annotated bubbles along with their topology type in the graph (path-explicit or alignment-rescued).

2.3 Genomic material

2.3.1 Simulated genome datasets

We generated several simulated genome sets with well-controlled inversion variants in order to investigate how inversions are represented in pangenome graphs of increasing complexity. We used

the genomic sequence of the chromosome 6 of a butterfly species (*Coenonympha arcania*) of length 20 Mb as first genome of each set (Legeai et al., 2024). We defined a set of 100 non-overlapping inversions, whose coordinates were randomly positioned on this chromosome, and whose sizes were picked uniformly between 50 bp and 1 Mb, for a total size of 14 Mb. We then simulated four synthetic chromosomes containing all these 100 inversions but with additional single nucleotide mutations uniformly introduced at different rates: 0, 1 and 5 % rate. As a result, we obtained three 2-genome sets each corresponding to a different nucleotidic divergence level and including the real chromosome sequence as reference and a synthetic genome with its 100 inversions.

We also produced a genome set containing 10 haplotypes, as pangenome graphs are more oftenly used to represent larger collections of genomes. This set contains the reference chromosome 6 along with 9 synthetic haplotypes at a 0% single nucleotide mutation rate. One of the syntetic haplotype contains the whole set of 100 synthetic inversions, while the 8 remaining ones each contain a random sample of 50 inversions among the 100.

2.3.2 Human dataset

For application on human genomes, we based our genome set construction on the extensive inversion dataset published by (Porubsky et al., 2022), reporting 292 balanced inversions accross the GRCh38 reference and genotyped in 44 individuals. We used the GRCh38 human reference genome, as well as the diploid genome assemblies of the four individuals NA19240, HG00733, HG03486 and HG02818 published by the HPRC (Liao et al., 2023), which were the only assemblies available among the 44 genotyped genomes. In order to limit the processing cost, we selected a single chromosome to process. We filtered the inversion dataset to remove unbalanced inversions (*i.e.* duplicated inversions) and low confidence inversion calls (without the 'PASS' tag). We finally selected the chromosome 7, of 159 Mb, which with 18 inversions present in the four selected genomes was the most inversion-rich autosome in this dataset.

We ran RagTag v2.1.0 (Alonge et al., 2022) to resolve the assemblies' contigs correspondance to GRCh38 chromosome 7, and extracted the corresponding (unscaffolded) contigs in a separate multi-fasta file for each of the 9 haplotypes.

2.3.3 *Coenonympha* butterfly dataset

For application on more divergent genomes, we constructed another dataset with the diploid assemblies of 4 closely related *Coenonympha* (Satyrinae) butterfly species, namely *Coenonympha arcania*, *Coenonympha gardetta*, *Coenonympha darwiniana* and *Coenonympha cephalidarwiniana*, between which 12 large inversions ($\geq$ 100 Kb) were discovered through whole genome alignment (Romain et al., prep). We added the haploid assembly of *C. arcania*, which was used as the reference against which the inversions were discovered, as the reference genome of this set, producing a genome set of 9 haplotypes. Among the 7 chromosomes of *C. arcania* bearing inversions, we selected the chromosomes 6, 10, 14, 21 and 26 for a total of 8 large inversions. We chose to exclude the Z sex chromosome and one autosome on which the inversion shows very high repetitive and complex genomic context to limit the complexity of the graphs. We extracted the 5 selected chromosomes in a separate multi-fasta file for each of the 8 haplotypes (*i.e.* two haplotypes per genome) and for the *C. arcania* reference haplotype of the species complex, for a total haplotype size of 83 Mb.

2.4 Construction of the pangenome graphs

For each set of genomes, we constructed four versions of pangenome graphs using Minigraph (Li et al., 2020), Minigraph-Cactus (Hickey et al., 2024), PGGB (Garrison et al., 2023) and Progressive Cactus (Armstrong et al., 2020) to explore the diversity of inversion representation across the state of the art tools.

We ran Minigraph-Cactus v2.8.2 with the parameters '–clip 0 –filter 0' to retain all haplotype paths in the graphs. PGGB v0.6.0, Minigraph v0.21 and Progressive Cactus v2.8.2 were run with default parameters. As it is recommended to use binarized input trees with the more recent versions of Progressive Cactus, we wrote arbitrary trees with uniform branch length to ensure they have the right structure, keeping haplotypes of single individuals closest together in the trees. We recon that the trees should have minimal impact on human and simulated data experiments as genomes are close, and we built the *Coenonympha* genome tree corresponding to Capblancq et al. (2019). We then converted the resulting HAL file of Progressive Cactus to a VG graph format using the *hal2vg* tool (included in Cactus v2.8.1) with the '–noAncestors' parameter. *Vg convert* (Garrison et al., 2018) v1.61, was then used with the '-f -W' parameters to obtain the Minigraph and Progressive Cactus graphs in GFA 1.0 format.

Variant bubbles were obtained using *vg deconstruct* or the *minigraph-call* pipeline, depending on the pangenome graph format. The reference genome, used as coordinate system for the output VCF file, was set as GRCh38 for the human dataset, *C. arcania* for the *Coenonympha* dataset, and the reference sequence for the simulated datasets. For PGGB and Minigraph-Cactus, this step was included in their pipeline. For Minigraph and Cactus, we ran *vg deconstruct* with the parameter '-a' to obtain an exhaustive set of bubbles (without any filter).

All command lines for graph construction and bubble calling pipelines are given in Supplementary Material.

2.5 Evaluation of inversion detection in the pangenome graphs

To assess the completeness of the sets of annotated inversion bubbles we calculate a recall value as the percentage of known inversions that were annotated in the graph by at least one of its bubbles. We established the correspondence of the annotated bubbles to the true inversions in the datasets based on their position overlap on the reference genome using bedtools (Quinlan, 2014) v2.27.1. We considered a given inversion as correctly annotated if a reciprocal overlap of at least 50% was found between the inversion and one of the annotated bubbles (*i.e.* the position of the annotated bubble on the reference covers $\geq 50\%$ of the size of the inversion, and vice-versa), using the options '-f 0.5 -r' of the function *bedtools intersect*. We also calculated the number of imprecise inversion bubbles, which we define as annotated bubbles that may overlap a true inversion but do not meet the 50% reciprocal overlap requirement.

3 Results

3.1 Investigating inversion bubble topologies in most simple simulated graphs

We first investigated how inversions are represented in pangenome graphs that we can imagine to be the easiest instances for inversion detection, that are graphs built from only two genomes differing

only by non overlapping and randomly positioned inversions. To do so, we built graphs for the set of two 20 Mb chromosomes differing by 100 inversions and without any other polymorphism (see Section 2.3.1), using four pangenome graph pipelines Minigraph, Minigraph-Cactus (MGC), Progressive Cactus (Cactus) and PGGB.

3.1.1 Pangenome graph complexity

For such a simple genome set, we expect its pangenome graph to contain few nodes and edges and a total sequence content of exactly 20 Mb, since inversions do not produce any novel sequences, only novel adjacencies. In theory, considering N inversions, we expect $1+2N$ nodes and $4N$ edges. In practice, we obtained rather different graph statistics for all four tools (Table 1). Except for Minigraph, all three other tools obtained more than ten times more nodes and edges than expected in their graphs, with the largest value for MGC with more than 200,000 nodes. Furthermore, only Cactus and PGGB have a graph size close to the expected 20 Mb. Minigraph and MGC graphs contain more than 5 Mb of additional sequences, being either duplicated or more probably present in both forward and reverse-complement versions in the graph.

Table 1: Graph and bubble statistics of the pangenome graphs obtained for the simulated genome set with 100 inversions and no punctual polymorphism, for each pangenome pipeline tested. The expected size of this graph is 20 Mb with 100 bubbles. The number of bubbles indicates the total number of bubbles reported by the pipeline bubble callers. Large bubbles are those with at least two paths of similar length greater than 50 bp. Inversion bubbles are those annotated as inversion by our method. The Recall column indicates the percentage of simulated inversions (among the 100) that have a significant reciprocal overlap with at least one inversion annotated bubble. ("MGC": Minigraph-Cactus).

PG pipeline	Graph size (Mb)	# Nodes	# Edges	# Bubbles	# Large bubbles	# Inversion bubbles	Recall (%)
Cactus	19.99	7,275	9,520	1,216	65	65	63
Minigraph	25.49	131	190	48	28	16	7
MGC	27.27	204,054	220,492	16,271	73	62	12
PGGB	20.21	3,589	4,947	1,298	112	90	88

3.1.2 Many inversions are not represented as bubbles in the graphs

We applied the standard bubble caller tools to find bubbles in these graphs, namely *vg deconstruct* for graphs built with MGC, PGGB and Cactus which are in GFA1 format (including the genome paths as P or W-lines), and the *minigraph-call pipeline* for the minigraph graph in rGFA format (without any P or W-line). Then we applied our method to look for inversion signals in these bubbles. We expected 100 bubbles in total in the graphs, as the two haplotypes used to construct the graphs differ by only 100 inversions. Minigraph graph contains half the number of expected bubbles, while Cactus, PGGB and MGC graphs contain respectively 10, 10 and 160 times more bubbles than expected (Table 1). For all graphs but PGGB, less than 100 bubbles pass our method's filter on bubble size (*i.e.* at least two paths $\geq$ 50 bp and of similar size), which represent 0.4, 5.3,

8.6 and 58 % of the total number of bubbles in the graphs of MGC, Cactus, PGGB and Minigraphs, respectively. Most of the large and balanced selected bubbles are annotated as inversions, though in fewer number than expected: respectively 16, 62, 65 and 90 for Minigraph, MGC, Cactus and PGGB graphs.

We compared the coordinates of annotated bubbles to the ones of the inversions that we simulated, and obtained even fewer percentages of simulated inversions that significantly overlapped annotated bubbles, with recall values of 88, 63, 12 and 7 % of with the PGGB, Cactus, MGC and Minigraph pipelines, respectively (Fig. 3A). It means that between 12 and 93 % of the simulated inversions were not overlapped by at least one inversion-annotated bubble with a reciprocal overlap of at least 50%. The size of the inversions did not seem to impact their bubble representation in Minigraph's graph. On the other hand, for all other three graphs, the inversions not found as bubbles are generally the smallest (less than 100 kb with MGC, less than 50 kb with Cactus and PGGB) (Fig. 3C).

All annotated bubbles overlap by at least one bp a true inversion. However, for all pipelines, the number of annotated bubbles is higher than the recall value (Tab. 1). There are two reasons for this difference: either the boundaries of some inversion bubbles are not sufficiently precise and these bubbles do not meet the 50% reciprocal overlap with the true inversion recall criterion, or there is a certain amount of redundancy in the annotated bubbles, with several bubbles nested within each other and each overlapping the same true inversion. In some cases, we even observe these two factors combined. These two reasons contribute differently to the difference in number of annotated bubbles and recall value for the different pipelines (see Supplementary Table 1). Minigraph and PGGB pipelines show mainly breakpoint precision issues, with some inversion bubbles being restricted to a too small portion of the true inversion, or on the contrary extending way beyond the true inversion breakpoints. An extreme example is observed in the Minigraph graph, where an inversion bubble of 443 Kb overlaps four consecutive true inversions (although separated by at least 14 Kb from one another). For Cactus and MGC pipelines, the main differences between the number of annotated bubbles and the recall is due to redundancy issues, with several multi-level bubbles (or 'nested' bubbles) at some inversion sites. One extreme case is observed in the MGC graph where a single true inversion of 5 Kb is overlapped by 22 distinct inversion bubbles (*i.e.* distinct variant lines in the VCF file). This redundancy issue was rather unexpected for such a simple genome set, with only two haplotypes with no other polymorphism than the 100 non-overlapping inversions.

3.1.3 Several inversion bubble topologies between graph pipelines

Bubbles annotated as inversions were divided in two types : (i) "path-explicit": the signals of inversion can be directly identified in the graph (ii) "alignment rescued": the inversion has been detected after alignment of the bubble path sequences. As a result, we observed that the graphs differ in the type of topology found for the inversion bubbles (Fig. 3B). All inversion bubbles detected in the Cactus graph are *path-explicit*, while there are both *path-explicit* and *alignment-rescued* topologies in the other graphs. Interestingly, we found that the inversion topology in PGGB is strongly linked to the inversion size, as all path-explicit inversion bubbles are larger than 50 kb, and all alignment-rescued ones are smaller than this threshold (Fig. 3C). In the case of MGC, the path-explicit inversions are also larger in size than the alignment-rescued ones (threshold around 500 kb). It seems to be the opposite for Minigraph, although the threshold is not that strict as the inversion size range between the two topology types overlaps around 100 kb.

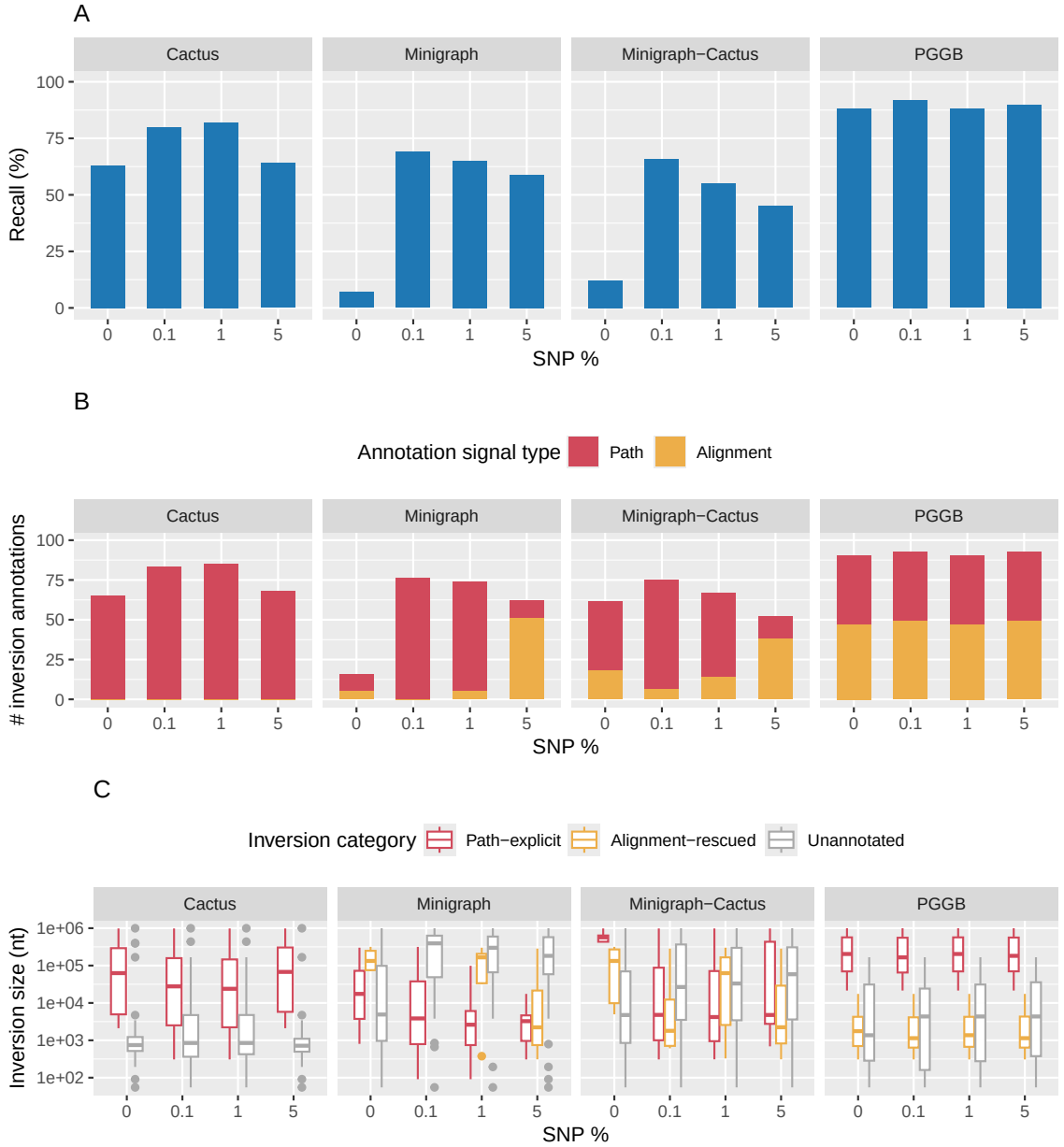

Figure 3: Inversion bubble statistics for the 2-haplotype simulated datasets pangenome graphs depending on the PG pipeline. (A) Recall of inversions as calculated by the percentage of simulated inversions overlapped by at least one bubble annotated as inversion. A minimum threshold of 50% reciprocal overlap was used. (B) Number of inversion bubbles based on a path signal ('path explicit', *red*) or an alignment signal ('alignment-rescued', *orange*). (C) Size distribution of the true inversions annotated by path signal (*red*), annotated by alignment signal (*orange*), or unannotated (*grey*).

3.1.4 Accuracy of inversion breakpoints

We measured the accuracy of the inversion breakpoints represented in the pangenome graphs by calculating the offset of both start and end positions of the true positive inversion annotated bubbles compared to the true positions of the inversions in the simulated datasets (Fig 4). An offset of 0 means that the breakpoints of the inversion-annotated bubble correspond exactly to that of the inversion positions in the genome, while a greater offset means that the inversion-annotated bubble extends before or after the inversion true positions in the genome (Fig 4).

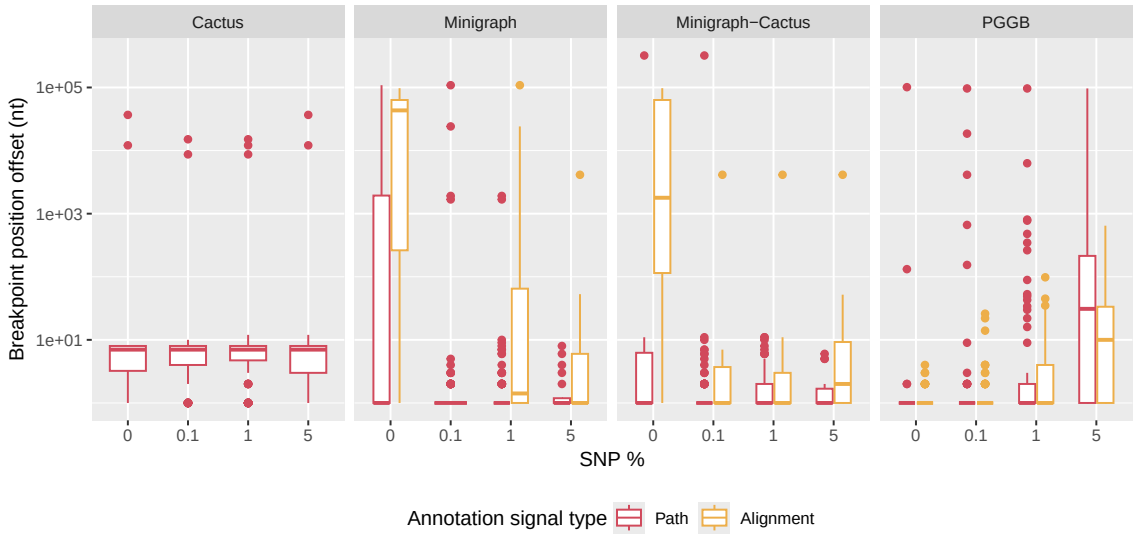

Figure 4: Inversion bubble position offsets in the pangenome graphs. The Y axis shows the position offset, which is computed as the sum for both breakpoints of absolute position difference between the inversion-annotated bubble coordinate and its corresponding true inversion coordinate in the reference genome. The Y axis is log-scaled.

Among all pangenome graphs on simulated datasets at 0% SNPs, the Cactus graph shows the most accurate annotated bubble positions with an offset mostly smaller than 10 bp. As expected, Minigraph appeared as the less precise tool concerning inversion coordinates. However, for almost all tools, we can observe a number of extreme offset values as large as dozens of Kbs. Interestingly, we found that in PGGB graphs, the path-explicit inversion bubbles show an overall greater position offset than the alignment-rescued ones, with differences enlarging the inversions, which means that the bubbles "capture" extra neighbouring sequence.

3.2 Factors impacting inversion bubble recovery in simulated graphs of increasing complexity

Next, we simulated more realistic and complex graphs by increasing the nucleotidic divergence of the input haplotypes and by increasing the number of haplotypes in the graphs. On one hand, we built graphs for the sets of two 20 Mb haplotypes differing both by 100 inversions and 0.1, 1 or 5 %

SNPs to test the impact of the single nucleotide divergence (see Section 2.3.1). On the other hand, we built graphs for the set of ten 20 Mb haplotypes, differing by varying subsamples of 100 initial inversions, and without any other polymorphism (see Section 2.3.1).

3.2.1 Increasing the level of single nucleotide divergence

As expected, the three pipelines aiming at representing the whole range of variants show a great increase in node, edge and bubble numbers in their graphs compared to the graphs obtained for genome sets without small polymorphisms (Tab. 2). These are in the same order of magnitude as the number of simulated punctual variants (20,000, 200,000 and 1 million for 0.1, 1 and 5 % divergence datasets respectively). Although Minigraph is not meant to represent small polymorphism, the presence of SNPs increases a little bit its graph complexity, with hundreds of additional nodes compared to the graph obtained for the genome set without punctual divergence. Surprisingly, Minigraph and MGC pipelines produced graphs of size closer to the haplotype size when the nucleotidic divergence increases.

Table 2: Graph and bubble statistics of the pangenome graphs obtained for the 2-haplotype simulated genome sets with 100 inversions and with varying percentages of punctual polymorphism, for each pangenome graph pipeline used. All of the genome sets have a haplotype size of 20 Mb. The number of bubbles indicates the total number of bubbles reported by the pipeline bubble caller. The Recall column indicates the percentage of simulated inversions (among the 100) that have a significant reciprocal overlap with at least one inversion annotated bubble. ("MGC": Minigraph-Cactus).

Genome set	PG pipeline	Graph size (Mb)	# Nodes (10^3)	# Edges (10^3)	# Bubbles	# Inversion bubbles	Recall (%)
0.1% SNP	Cactus	20.02	64.7	86.1	21,019	83	80
	Minigraph	20.87	0.2	0.4	106	76	69
	MGC	22.34	295.1	385.4	90,126	75	66
	PGGB	20.24	63.2	84.5	21,023	93	92
1% SNP	Cactus	20.20	601.2	800.8	198,952	85	82
	Minigraph	21.43	0.3	0.4	107	74	65
	MGC	23.92	864.1	1,144.1	279,837	67	55
	PGGB	20.45	587.6	783.5	195,771	90	88
5% SNP	Cactus	21.00	2,902.8	3,853.9	950,032	68	64
	Minigraph	21.39	0.3	0.4	89	62	59
	MGC	25.28	2,338.7	3,101.3	762,558	52	45
	PGGB	21.58	2,718.8	3,625.4	906,601	93	90

Even more surprising is the sharp increase of inversion recall with the Minigraph and MGC graphs at nucleotidic divergence of 0.1 % compared to nucleotidic divergence of 0 %, rising from 7% to 69% of simulated inversions found as inversion bubbles for Minigraph and from 12% to 66% for MGC (Fig. 3A). The increase of nucleotidic divergence from 1 to 5 % appears to impact the recall, in particular for the Cactus and MGC graphs for which the recall drops by 18 and 10 % points,

respectively. We still obtain the highest inversion recall values with the PGGB graphs (88-92 %), regardless of the punctual nucleotidic divergence.

The increase of nucleotidic divergence impacts the level of redundancy of annotated bubbles for the MGC pipeline, which was very high for the 0% divergence simulation and become negligeable for high divergences (Supplementary Table 1). On the contrary, more than 57 % of the inversion bubbles of the Cactus pipeline remain redundant with one another. While the topology of inversion bubbles seems unrelated to nucleotidic divergence for Cactus and PGGB pipelines, we observe a switch from a majority of path-explicit topology types at 0.1 and 1 % divergence to a majority of alignment-rescued topology types at 5 % divergence for Minigraph and MGC pipelines (Fig. 3B). Notably, the inversion breakpoint position offset for Minigraph and MGC pipelines is lowered for all nucleotidic divergence levels compared to 0 %, and becomes stable under 10 bp for MGC pipeline (Fig 4). On the other hand, the increase of nucleotidic divergence from 0 to 5 % appears to increase the overall imprecision of breakpoint positions for the PGGB pipeline, while it remains stable and mainly under 10 bp for the Cactus pipeline.

3.2.2 Increasing the number of haplotypes in the graph

When increasing the number of haplotypes in the graph from two to ten in a 0 % nucleotidic polymorphism context, we observe an increase in the number of simulated inversions that can be found as bubbles with inversion signal in all graphs, although the number of simulated inversions is not changed (Tab. 3). The highest gains in recall are observed for Cactus (+ 12 % points) and MGC (+ 10 % points) pipelines. However, not all 100 inversions could be found as inversion bubbles in any of the graphs, and ten inversions were still missing in the best tool, PGGB. Additionnally, increasing the number of haplotypes in the graphs doubled the number of bubbles annotated as inversions for the Cactus pipeline, from 65 to 140 annotated bubbles, most of them being redundant with other annotated bubbles (see Supplementary Table 2).

Table 3: Inversion annotation recall obtained for the 2-haplotype and 10-haplotype simulated genome set with no punctual polymorphism, for each pangenome graph pipeline used ("MGC": Minigraph-Cactus).

PG pipeline	Recall (%)	
	2 haplotypes	10 haplotypes
Cactus	63	75
Minigraph	7	10
Minigraph-Cactus	12	22
PGGB	88	90

3.3 Inversion bubble annotation in real pangenome graphs

Finally, we investigated how real inversion variants are represented in pangenome graphs constructed with two real genome datasets. The first dataset used is a human genome dataset containing nine haplotypes for the 159 Mb chromosome 7, for which we would expect to find at least 18 inversions described in Porubsky et al. (2022) (see Section 2.3.2) ranging from 1.0 kb to 1.7 Mb, with a median size of 41.2 kb. In order to explore the inversion representation in pangenome graphs of more divergent genome sets, we designed another dataset from four species of alpine *Coenonympha*

butterfly. While these species are highly related and interfertile, the nucleotidic divergence between their genome is estimated to be around 7% using Mash (Ondov et al., 2016). The four genomes were recently assembled, and 12 large inversions ($\geq$ 100 kb) were identified by pairwise whole genome alignment (Romain et al., prep). For this dataset, we selected 8 large inversions (see Section 2.3.3) ranging from 106 kb to 1.7 Mb, with a median size of 511 kb, and spread over 5 autosomes, amounting to a total haplotype size of 83 Mb.

As expected, we obtained graphs of a much higher complexity than for simulated genomes, illustrated in Table 4. Notably, the *Coenonympha* graphs have a striking total size of 2.8 to 4 times the single haplotype size, while the size of the human graphs is of the same order of magnitude than that of the single haplotype. We obtained a very low recall of inversion annotation for both datasets, no more than 27.7% and 12.5% of the expected inversions were found among the called bubbles, for the human and *Coenonympha* datasets respectively (Tab. 4). Although the number of inversion annotated bubbles in the *Coenonympha* graphs exceeds the number of known inversions in the dataset by 3 to 4 fold, only 2 of these bubbles overlap one of the known inversions in each of the Minigraph, MGC and PGGB graphs, and all but one (in the PGGB graph) have a size an order of magnitude smaller than the expected size of the inversions and thus do not pass the 50% reciprocal overlap requirement of the recall.

Table 4: Graph and bubble statistics of the pangenome graphs obtained for the human and *Coenonympha* genome sets, for each pangenome graph pipeline ("MGC": Minigraph-Cactus). The human genome set has a haplotype size of 159 Mb, the *Coenonympha* genome set has a haplotype size of 83 Mb.

Genome set	PG pipeline	Graph size (Mb)	# Bubbles (10^3)	# Inversion bubbles	Recall (%)
Human, chr7 (9 haplotypes)	Cactus	176.87	815.5	6	22.2
	Minigraph	162.04	2.9	6	22.2
	MGC	178.85	657.3	36	27.7
	PGGB	168.10	628.3	3	11.1
Coenonympha (9 haplotypes)	Cactus	333.71	6,135.3	37	0.0
	Minigraph	228.84	63.5	25	0.0
	MGC	321.32	4,750.7	29	0.0
	PGGB	288.24	4,235.1	32	12.5

The running time and memory requirement of INVPG-annot on the pangenome graphs VCFs constructed on the human dataset and with the four state of the art pipelines are shown in Table 5. INVPG-annot ran in less than one minute on the Cactus and Minigraph VCFs, and in 11 minutes on the MGC VCF, while it took more than two hours to run on the PGGB VCF. We observed that this running time is correlated to the number of allele alignments performed, which is with little surprise the limiting factor of our tool's running speed. We observed the same correlation for the memory consumption of INVPG-annot.

Table 5: Total CPU time and maximum memory requirement of inversion annotation from the VCF files obtained by all pipelines' bubble callers from the human pangenome graphs. All annotations were run on 1 CPU thread.

PG pipeline	Total CPU time (min)	Max memory (Mo)
Cactus	0.6	1.2
Minigraph	0.2	1.2
Minigraph-Cactus	11.0	533.8
PGGB	146.2	1,946.5

4 Discussion

4.1 Very contrasted inversion handling between the pangenome graph pipelines

In this work, we investigated how large inversion variants are represented in pangenome graphs and more precisely whether we can find them in graph bubbles, which is the classical graph pattern searched for when calling and analyzing genomic variants in a pangenome graph. Our results on very simple simulated datasets highlighted highly contrasted representations of inversion variants between the four tested pangenome pipelines, namely Cactus, Minigraph, MGC and PGGB. The PGGB pipeline reports more inversion bubbles, with a stable recall around 90 % for all simulated datasets. On the contrary, the number of inversions recovered by Minigraph and MGC pipelines is much more variable, depending on the punctual divergence in simulated datasets. In particular, in the non-realistic case of no punctual polymorphism, their recall is strikingly low, around 10 %. In more realistic simulations, still only 55 to 65 % of inversions were found as bubbles in these graphs. As the inversion signals required to annotate bubbles as inversions are not very stringent (at least 50 % of the bubble's reference allele comprised in segments with opposite path traversals or in reverse alignments), we hypothesize that some missed inversions are not fully contained in input bubbles. In particular in the case of MGC, we suspect that inversion loci that are firstly represented as unrelated sequences at the Minigraph step could then be split at the Cactus step and end up represented as chains of bubbles in the graph, thus spread over multiple independent bubbles in the VCF. It could also be that some specific bubble topologies in the graph are not reported by the bubble detection tool. But, after eliminating the inversion size hypothesis, we found no other simple inversion characteristic to explain the absence of some of the bubbles. Unfortunately, we could not find sufficient details on bubble detection algorithms to guide these investigations. A possible improvement of the method would therefore be to directly scan the graph for the target topologies in order to bypass the constraints of the bubble detection tools.

The pangenome graph pipelines also showed different patterns of inversion topology distribution, and the need to re-align bubble alleles proved to be especially relevant for inversions of smaller size (below 50 kb) in the PGGB graphs. We found that the inversion size threshold between path-explicit and alignment-rescued topologies seems to be mainly dependant on the parameter set used for the genome alignment step in PGGB with wfmash (‘-s‘, ‘-l‘ and ‘-c‘). When wfmash outputs large alignments encompassing smaller inversions, these inversions are not detected by seqwish which analyses only CIGAR strings to find sequence variation. After testing different wfmash parameters values, we found that the default values of ‘-s‘, ‘-l‘ and ‘-c‘ were those giving the best inversion

recall with the simulated datasets.

4.2 The issue with inversions represented as unrelated sequences

While inversions represented as unrelated sequences in the graph (*i.e.* with distinct nodes for the ancestral and inverted versions of the sequence) can still be recovered if there is at least one bubble containing the entire locus, they need additional alignment to be annotated, which can become a computational burden for more large and complex graphs (more than 2 hours for the annotation step only with the PGGB human pangenome graph's bubbles). Moreover, those sequences with a same origin and position are represented twice in the graph, which could raise issues with downstream processing and analyses of the graph. For example, if there are small variants (e.g. SNPs) inside such inversions that one wishes to genotype, the question arises as of how these are represented and detected (on one or both of the inversion distinct paths, or simply missed) and how accurate the mapping of reads would be between those distinct paths. Reads originating from the inverted locus could be mapped to either of the paths, generating multi-mapping problems. As a conclusion, these inversions that are not explicitely represented in the graph need to be recovered as a path-explicit topology to reduce the redundancy and ambiguity of the pangenome graphs, otherwise limiting their usage.

4.3 A new tool to annotate inversions and inspect their topology in pangenome graphs

In addition to the comparison of pangenome graph pipeline's behaviour with respect to inversion variants, we presented a method and tool which detects inversions in pangenome graphs through the analysis of variant bubbles such as those output by *vg deconstruct* and *minigraph-call*. Its stringency on inversion allele size variation can be adapted to the diversity of the pangenomes, and the stringency of required coverage for inversion signal can also be set to the desired value, although these parameters can affect both computing time and annotation precision. With a larger inversion size variation and a graph rich in unrelated-sequence-like bubbles, the amount of alignment performed could slow the annotation process. While a lower stringency on required coverage of inversion signal helps recover inversions "hidden" in larger bubbles, it can also decrease the precision of the annotation. One way to improve the detection of these inversion could be to analyze the distribution of inversion signals along the bubble paths in order to refine the inversion breakpoint positions. Furthermore, with the provided type of topology for each detected inversion, we argue that it could be used to eventually perform targeted modifications of the pangenome graph in order to transform alignment-rescued inversion bubbles into path-explicit ones, thereby improving the quality of the pangenome graph.

5 Acknowledgements

We are thankful to the GenOuest bioinformatics platform for supporting the calculations. This work was supported by the French National Research Agency (ANR-20-CE02-0017).

References

Alonge, M., Lebeigle, L., Kirsche, M., Jenike, K., Ou, S., Aganezov, S., Wang, X., Lippman, Z. B., Schatz, M. C., and Soyk, S. (2022). Automated assembly scaffolding using ragtag elevates a new tomato system for high-throughput genome editing. *Genome biology*, 23(1):258.

Armstrong, J., Hickey, G., Diekhans, M., Fiddes, I. T., Novak, A. M., Deran, A., Fang, Q., Xie, D., Feng, S., Stiller, J., et al. (2020). Progressive cactus is a multiple-genome aligner for the thousand-genome era. *Nature*, 587(7833):246–251.

Berdan, E. L., Barton, N. H., Butlin, R., Charlesworth, B., Faria, R., Fragata, I., Gilbert, K. J., Jay, P., Kapun, M., Lotterhos, K. E., et al. (2023). How chromosomal inversions reorient the evolutionary process. *Journal of evolutionary biology*, 36(12):1761–1782.

Capblancq, T., Mavárez, J., Rioux, D., and Després, L. (2019). Speciation with gene flow: evidence from a complex of alpine butterflies (coenonympha, satyridae). *Ecology and evolution*, 9(11):6444–6457.

Cochetel, N., Minio, A., Guarracino, A., Garcia, J. F., Figueroa-Balderas, R., Massonnet, M., Kasuga, T., Londo, J. P., Garrison, E., Gaut, B. S., et al. (2023). A super-pangenome of the north american wild grape species. *Genome Biology*, 24(1):290.

Dabbaghie, F., Ebler, J., and Marschall, T. (2022). BubbleGun: enumerating bubbles and superbubbles in genome graphs. *Bioinformatics*, 38(17):4217–4219.

Garrison, E., Guarracino, A., Heumos, S., Villani, F., Bao, Z., Tattini, L., Hagmann, J., Vorbrugg, S., Marco-Sola, S., Kubica, C., et al. (2023). Building pangenome graphs. *bioRxiv*, pages 2023–04.

Garrison, E., Kronenberg, Z. N., Dawson, E. T., Pedersen, B. S., and Prins, P. (2022). A spectrum of free software tools for processing the vcf variant call format: vcflib, bio-vcf, cyvcf2, hts-nim and slivar. *PLoS Computational Biology*, 18(5):e1009123.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., et al. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36(9):875–879.

Hickey, G., Monlong, J., Ebler, J., Novak, A. M., Eizenga, J. M., Gao, Y., Marschall, T., Li, H., and Paten, B. (2024). Pangenome graph construction from genome alignments with minigraph-cactus. *Nature biotechnology*, 42(4):663–673.

Huang, K. and Rieseberg, L. H. (2020). Frequency, origins, and evolutionary role of chromosomal inversions in plants. *Frontiers in plant science*, 11:296.

Kirkpatrick, M. (2010). How and why chromosome inversions evolve. *PLoS biology*, 8(9):e1000501.

Legeai, F., Romain, S., Capblancq, T., Doniol-Valcroze, P., Joron, M., Lemaitre, C., and Després, L. (2024). Chromosome-Level Assembly and Annotation of the Pearly Heath Coenonympha arcania Butterfly Genome. *Genome Biology and Evolution*, 16(3):evae055.

Leonard, A. S., Crysnanto, D., Mapel, X. M., Bhati, M., and Pausch, H. (2023). Graph construction method impacts variation representation and analyses in a bovine super-pangenome. *Genome Biology*, 24(1):124.

Li, H. (2019). gfatools. `https://github.com/lh3/gfatools`.

Li, H., Feng, X., and Chu, C. (2020). The design and construction of reference pangenome graphs with minigraph. *Genome biology*, 21:1–19.

Liao, W.-W., Asri, M., Ebler, J., Doerr, D., Haukness, M., Hickey, G., Lu, S., Lucas, J. K., Monlong, J., Abel, H. J., et al. (2023). A draft human pangenome reference. *Nature*, 617(7960):312–324.

Liu, Y. H., Luo, C., Golding, S. G., Ioffe, J. B., and Zhou, X. M. (2024). Tradeoffs in alignment and assembly-based methods for structural variant detection with long-read sequencing data. *Nature Communications*, 15(1).

Mwaniki, N., Garrison, E., and Pisanti, N. (2024). Popping bubbles in pangenome graphs. *arXiv*.

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: fast genome and metagenome distance estimation using minhash. *Genome biology*, 17:1–14.

Porubsky, D., Höps, W., Ashraf, H., Hsieh, P., Rodriguez-Martin, B., Yilmaz, F., Ebler, J., Hallast, P., Maggiolini, F. A. M., Harvey, W. T., et al. (2022). Recurrent inversion polymorphisms in humans associate with genetic instability and genomic disorders. *Cell*, 185(11):1986–2005.

Quinlan, A. R. (2014). Bedtools: the swiss-army tool for genome feature analysis. *Current protocols in bioinformatics*, 47(1):11–12.

Romain, S., Capblancq, T., Després, L., Joron, M., Lemaitre, C., and Legeai, F. (in prep.). Characterization of large inversions to investigate hybrid speciation in the four species-complex of alpine coenonympha butterfly.

Sanders, A. D., Hills, M., Porubský, D., Guryev, V., Falconer, E., and Lansdorp, P. M. (2016). Characterizing polymorphic inversions in human genomes by single-cell sequencing. *Genome Research*, 26(11):1575–1587.

Wellenreuther, M. and Bernatchez, L. (2018). Eco-evolutionary genomics of chromosomal inversions. *Trends in ecology & evolution*, 33(6):427–440.