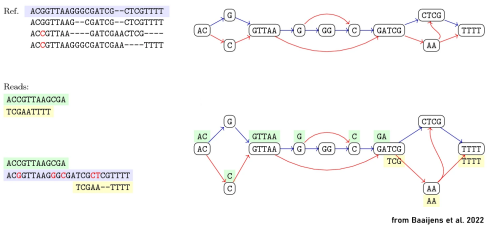
Towards an edit distance between pangenome graphs

Siegfried Dubois[1,3], Benjamin Linard[2], Matthias Zytnicki[2], Claire Lemaitre[1], Thomas Faraut[3]

1. Univ Rennes, Inria, CNRS, IRISA, Rennes, F-35000, France
2. MIAT, Université de Toulouse, INRAE, 31320 Castanet-Tolosan, France
3. GenPhySE, Université de Toulouse, INRAE, ENVT, 31320 Castanet-Tolosan, France

SeqBIM - 20$^{\text{th}}$ november 2023

Towards a pangenomic era ?



A variation graph :

- contains multiples genomes at once
- stores raw DNA sequences
- paths are genomes and variations

Replacing the reference genome

- allows for higher quality mapping [Eizenga et al. 2020]
- better genotyping of variants [Hickey et al. 2020]

Build a variation graph

From a variant set and a reference

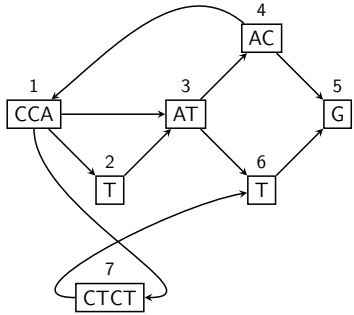
- Variation Graph toolkit (vg) [Hickey et al. 2020]

From a reference and a set of genomes

- minigraph (MG) [Li et al. 2020]
- minigraph-cactus (MGC) [Hickey et al. 2023]

From a set of genomes

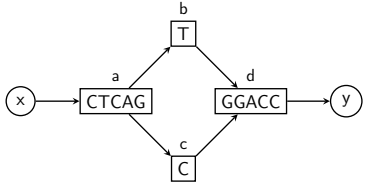
- PanGenome Graph Builder (PGGB) [Garrison et al. 2023]

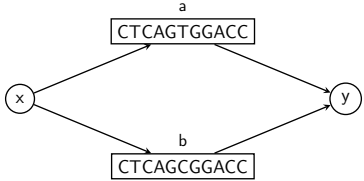


Problem

Graphs obtained from different *state-of-the-art* tools are different :

- number of nodes, edges... are different with the same input data [Leonard et al. 2023, Liao et al. 2023]
- no metric to compare them, nor to locate where the differences are





Definition of a variation graph

A graph $G = (V, E)$ represents a set of genomes $\Gamma = \{\Gamma_0, \Gamma_1, \ldots \Gamma_n\}$:

- each node $u \in V$ is associated to a string (or its *reverse-complement*) which is in at least one genome
- each arc $e \in E$ links two nodes which strings are contiguous in at least one genome and conveys the reading direction

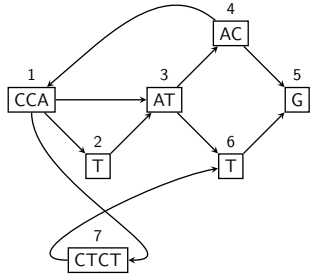


Figure 1 – Variation graph

Definition of a pangenome graph

A graph $G = (V, E, P)$ can be extended by a set $P = \{P_1, P_2 \dots P_n\}$ of paths :

- ordered and oriented list of nodes in the graph
- segmentation of a single embedded genome

We can say G expresses Γ_i if a path P_i is a segmentation of Γ_i

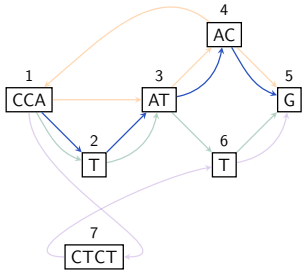


Figure 2 – Paths in a pangenome graph

$P_1 = 1^+, 2^+, 3^+, 4^+, 5^+$

CCA,T,AT,AC,G

$P_2 = 1^+, 2^+, 3^+, 6^+, 5^+$

CCA,T,AT,T,G

$P_3 = 1^+, 3^+, 4^+, 1^+, 3^+, 4^+, 5^+$

CCA,AT,AC,CCA,AT,AC,G

$P_4 = 1^+, 7^-, 6^+, 5^+$

CCA,AGAG,T,G

Complete pangenome graph

We will say that a graph $G = (V, E, P)$ is a **complete pangenome graph** if :

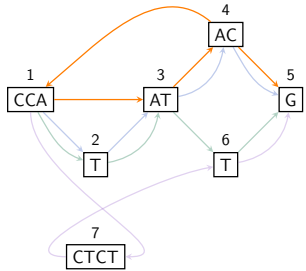
- the graph has a set P of paths
- there's one path per genome ($|P| = |\Gamma|$)
- the graph expresses Γ

S_1 = CCATATACG

S_2 = CCATATTG

S_3 = CCAATACCCAATACG

S_4 = CCAAGAGTG



$P_1 = 1^+, 2^+, 3^+, 4^+, 5^+$

CCA,T,AT,AC,G

$P_2 = 1^+, 2^+, 3^+, 6^+, 5^+$

CCA,T,AT,T,G

$P_3 = 1^+, 3^+, 4^+, 1^+, 3^+, 4^+, 5^+$

CCA,AT,AC,CCA,AT,AC,G

$P_4 = 1^+, 7^-, 6^+, 5^+$

CCA,AGAG,T,G

Figure 3 – Complete pangenome graph

Idea of our distance

We want to compare two graphs :

- A graph induces a segmentation for each genome
- A difference in segmentation implies different nodes

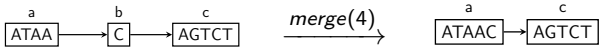
For each genome, we will simultaneously go through both segmentations

- Prevents an all-against-all comparison of the nodes between the two graphs

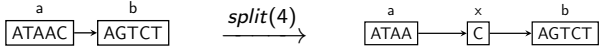
Editions on segmentation

Merges and splits are **editions** on a **segmentation** of a genome.

- **merge(x)** removes the breakpoint at the x^{th} genome position



- **split(x)** adds a breakpoint at the x^{th} genome position



For a genome with two segmentations, there exists a pair of sets M (merges) and S (splits) which allows to transform one segmentation to another. The union of the pair is an edition script.

Definition of our distance

Let $A = (V^A, E^A, P^A)$ and $B = (V^B, E^B, P^B)$ be two complete pangenome graphs with a shared set of genomes Γ.

The segmentation distance of the genome Γ_i will be :

$$d_s(P_i^A, P_i^B) = min(|M| + |S|) \tag{1}$$

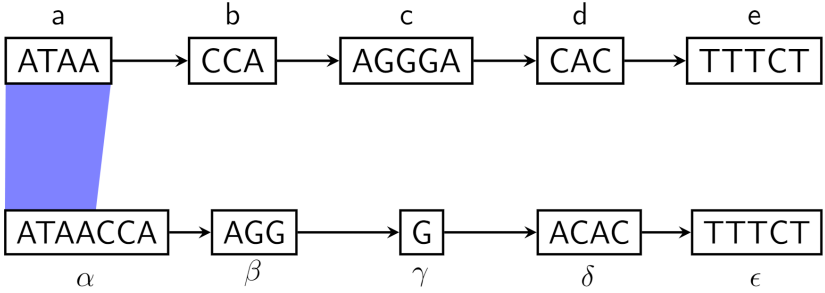
The distance is the sum of the segmentation distances of each genome :

$$d(A, B) = \sum_{i=1}^{|P|} d_s(P_i^A, P_i^B) \tag{2}$$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

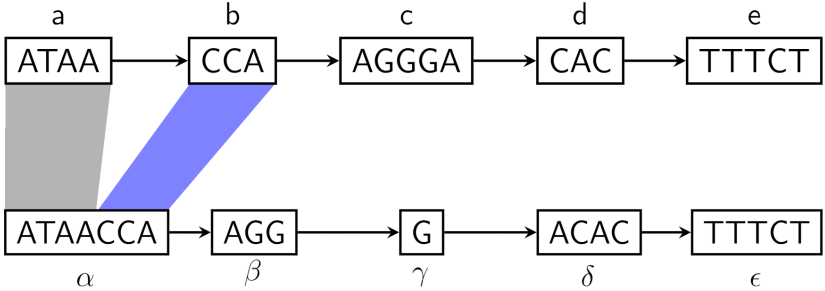


$M = \{\emptyset\}, S = \{\emptyset\}, i = 0, j = 0, p = [1, 4)$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

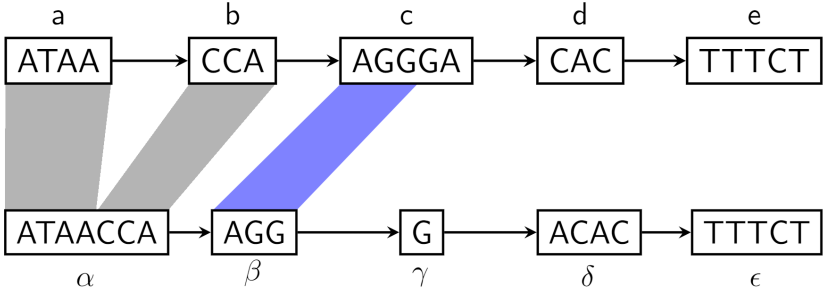


$M = \{4\}, S = \{\emptyset\}, i = 1, j = 0, p = [4, 7)$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

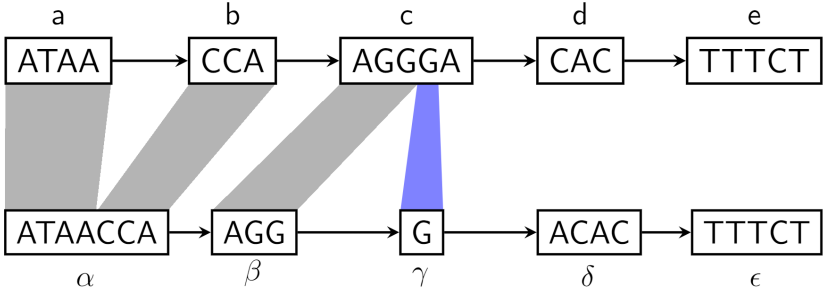


$M = \{4\}, S = \{\emptyset\}, i = 2, j = 1, p = [7, 10)$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

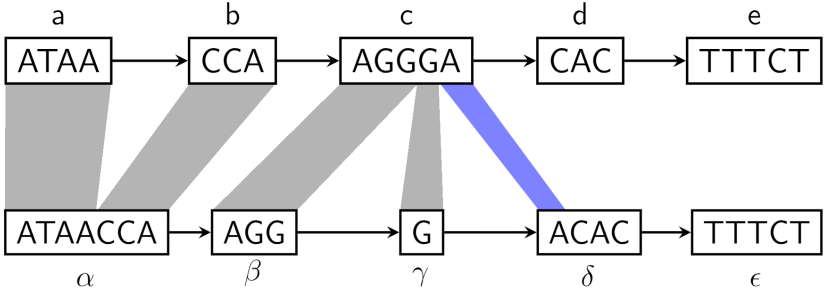


$M = \{4\}, S = \{10\}, i = 2, j = 2, p = [10, 11)$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

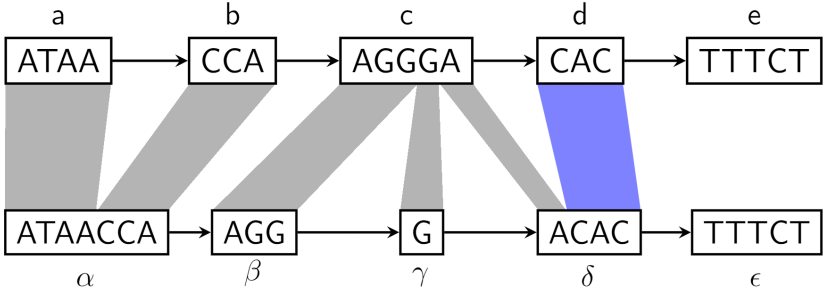


$M = \{4\}, S = \{10, 11\}, i = 2, j = 3, p = [11, 12)$

Algorithm

We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at

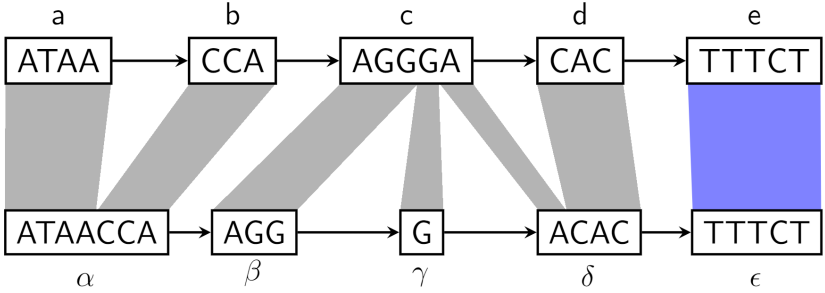


$M = \{4, 12\}, S = \{10, 11\}, i = 3, j = 3, p = [12, 15)$

Algorithm

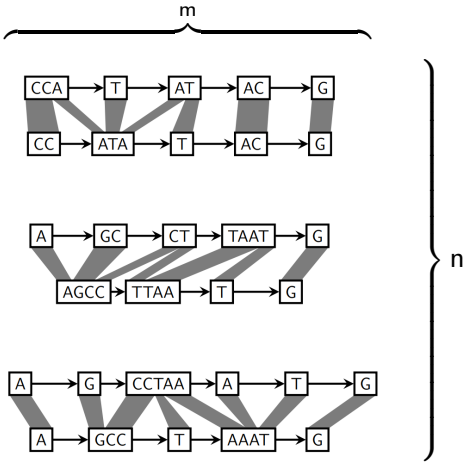
We compare the two segmentations while we maintain three variables :

- i and j current breakpoint indexes in each segmentation
- p current interval we are looking at



$M = \{4, 12\}, S = \{10, 11\}, i = 4, j = 4, p = [15, 20)$

Complexity



$$O(n \times m)$$

With n being the number of genomes and m the length of the genome

PANCAT

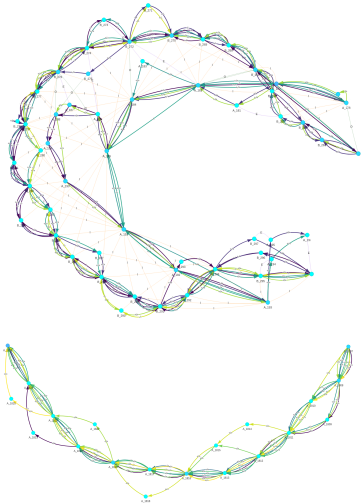
We implemented this algorithm in a Python tool, along with :

- diverse tools for variation graphs
- visualization that displays graphs side-to-side
- library to read, write and edit variation graphs

Regarding distance calculation :

- yeast data (200kb, 15 genomes, 30-50k nodes) : 9 sec. average
- maize data (20Mb, 4 genomes, 0.7-1.2M nodes) : 80 sec. average

https://github.com/Tharos-ux/pancat



Data

We build graphs from yeast assemblies [O'Donnell et al. 2023] with *MGC* and *PGGB*

- assemblies for the chromosome 1 of 15 individuals (200kb)
- both tools with default parameters
- multiple graphs for *MGC*, changing genome orders
- one graph for *PGGB*, which is *reference-free*

PGGB MGC

Figure 4 – Visualization with Bandage [Wick et al. 2015]

Distance between MGC graphs

Changing the reference in MGC has a greater impact than changing the order of inclusion of other sequences.

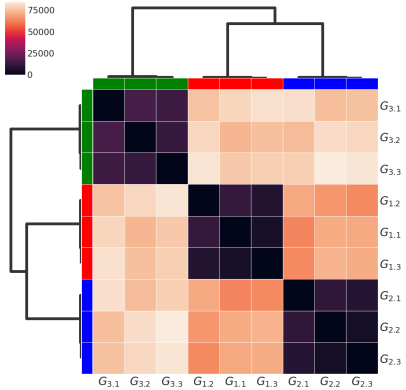


Figure 5 – Green, blue and red markers denotes a shared reference.

Distance between PGGB and MGC graphs

Choosing another sequence as reference with MGC can have more impact than switching to PGGB. The choice of the reference with MGC is significant !

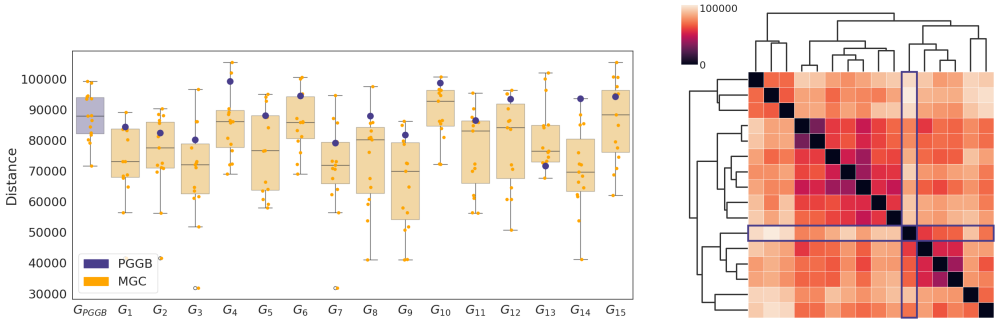


Figure 6 – Blue dots are distances to the PGGB graph, highlighted in blue on the clustermap.

Visualization

Exploring differences in segmentation for each node allows to view how graphs match

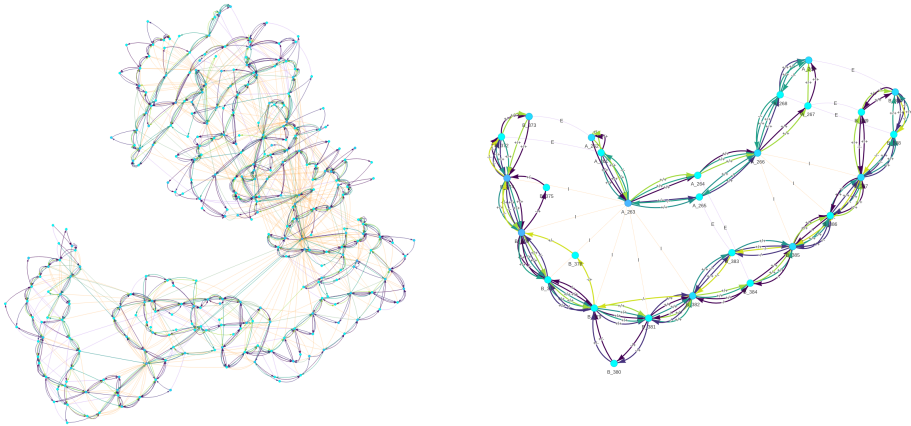


Figure 7 – Comparison of parts of PGGB and MGC graphs build with yeast data

Local events

Editions are not evenly distributed along the graph, but restrained to small areas.

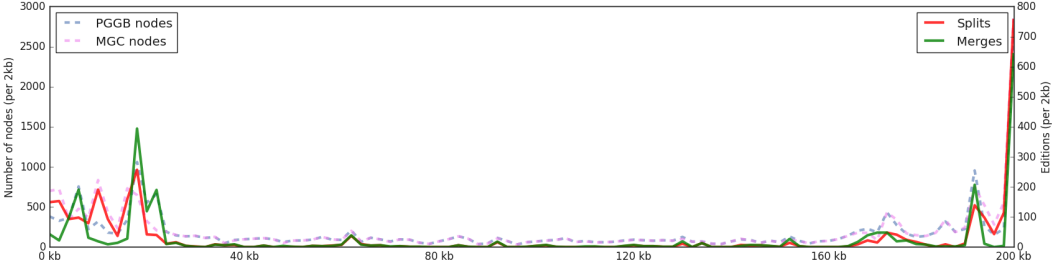


Figure 8 – This figure displays editions on a single genome, between two segmentations.

Contributions

We defined a distance between pangenome graphs :

- measure that relies on genome segmentation
- possible to pinpoint differences between graphs

We proposed an implementation :

- developped in Python
- a series of utilities, including a visualization tool
- with a library to parse variation graph formats

Perspectives

Distance is computed at *path-level* and not *graph-level*

- Loops does not count as differences
- The same operation on multiple paths is accounted multiple times
- Complexity can be lowered to $O(m)$, with m being the number of nodes

Future applications

- Compare edition results with variants data
- Explore large-scale graphs (ex : human pangenome [Liao et al. 2023])

Thanks for your attention !

We acknowledge the GenOuest bioinformatics core facility for providing the computing infrastructure.

PhD founded by the Agroecology and digital technology program.

The tool, **pancat**, is available here : https://github.com/Tharos-ux/pancat
A library **gfagraphs** for GFA format is here : https://pypi.org/project/gfagraphs