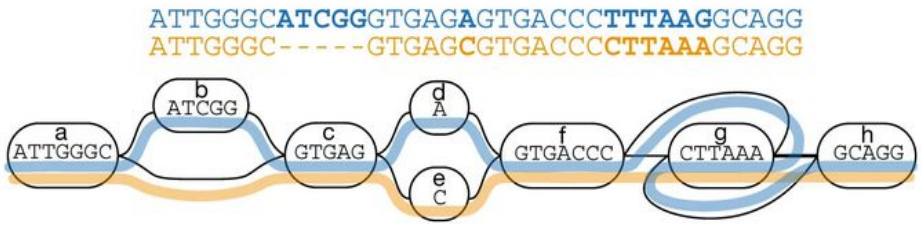
Methods for Interfacing with Graphs of Genomic Sequences

Pangenome graph manipulation for local visualisation with pancat

Siegfried Dubois, Thomas Faraut, Matthias Zytnicki and Claire Lemaitre

Lille - 23.09.2024

I. What is a pangenome graph?




```
S    a    ATTGGGC
S    b    ATCGG
…    …    ……
S    h    GCAGG
E    a    +    b    +    *
E    a    +    c    +    *
…    …    …    …    …    …
E    g    -    h    +    *
P    blue      a+,b+,c+... f+,g-,h+
P    orange    a+,c+,e+... f+,g+,h+
```

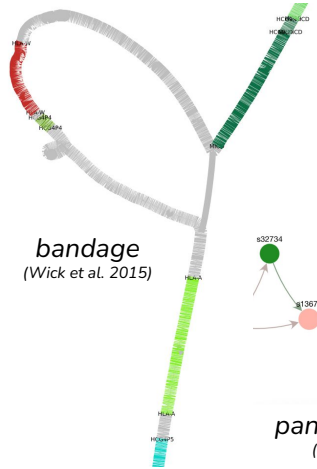
A structure among others to interface genome alignments

- Nodes are labelled with oriented genomic fragments
- Edges are adjacencies and paths represents haplotypes
- GFA is one format among others but is the most used

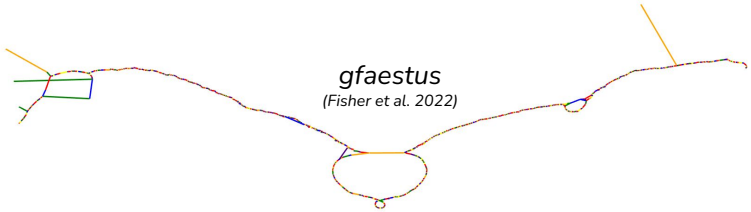
Type	Description
#	Comment
H	Header
S	Segment
L	Link
J	Jump (since v1.2)
C	Containment
P	Path
W	Walk (since v1.1)

Type	Regexp
A	`[!-~]`
i	`[-+]?[0-9]+`
f	`[-+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?`
Z	`[ !-~]+`
J	`[ !-~]+`
H	`[0-9A-F]+`
B	`[cCsSiIf](,[-+]?[0-9]*\.?[0-9]+([eE][-+]?[0-9]+)?)+`

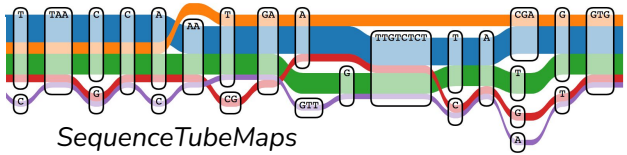
I. Pangenome graph visualisers



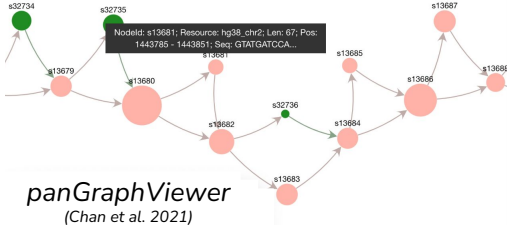
bandage
(Wick et al. 2015)



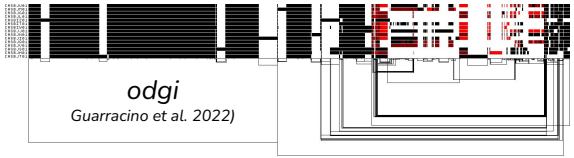
gfaestus
(Fisher et al. 2022)



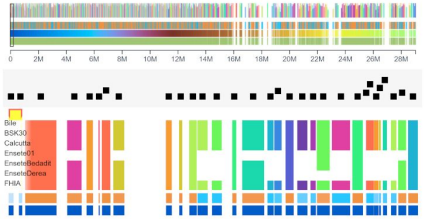
SequenceTubeMaps
(Beyer et al. 2019)



panGraphViewer
(Chan et al. 2021)



odgi
Guarracino et al. 2022)



panache
(Durant et al. 2021)

Many tools available, both static and dynamic ones

- Most of them take GFA as input format
  - “Graph view” (with a layout engine)
  - “Alignment view” (with blocks for homologies)

colindaven/**awesome-pangenomes**

A list of software for pangenomics

5 Contributors · 0 Issues · 62 Stars · 12 Forks

A repository that tracks tools to visualise and manipulate pangenomes

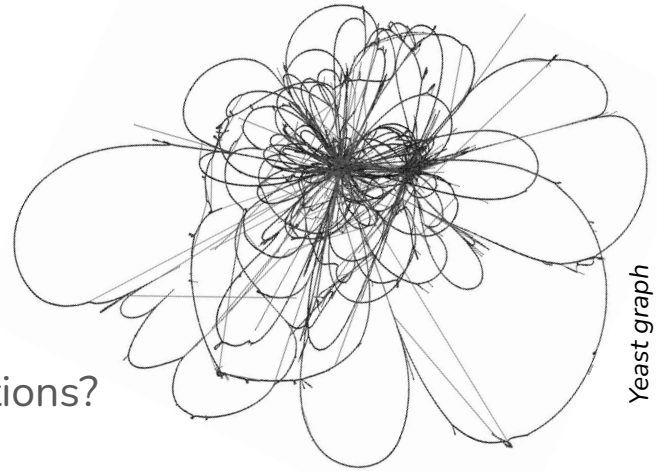
II. How to address visualisation

Objects:

- Huge structures with countless nodes
- Can be represented as bi-directed graphs
- Contains paths that represents haplotypes

Observations:

- A pangenome graph is huge and not easily read
- Different representations for different biological questions?



Yeast graph viewed with odgi

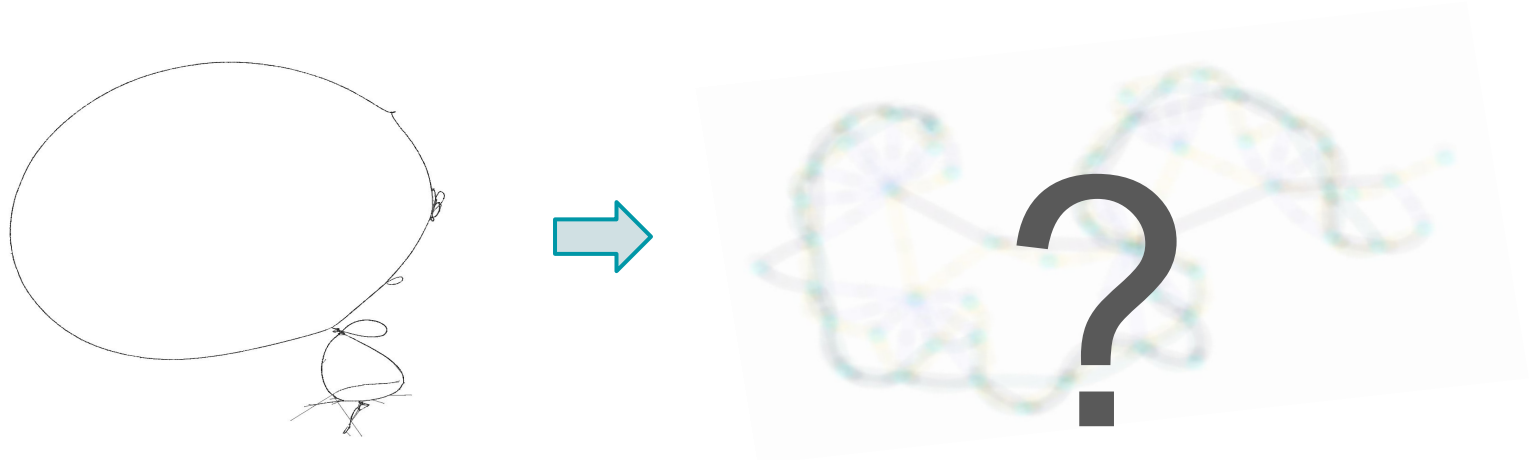
Species	*Mean genome size*	*Number of haplotypes*	*Number of nodes*
Yeast	12.07 Mb	42	1M
Bovine *(Leonard et al.)*	2.7 Gb	23	190M
Human *(Liao et al.)*	3.43 Gb	96	92M
Wild grape *(Cochetel et al.)*	1.12 Gb	18	200M

II. Our approach

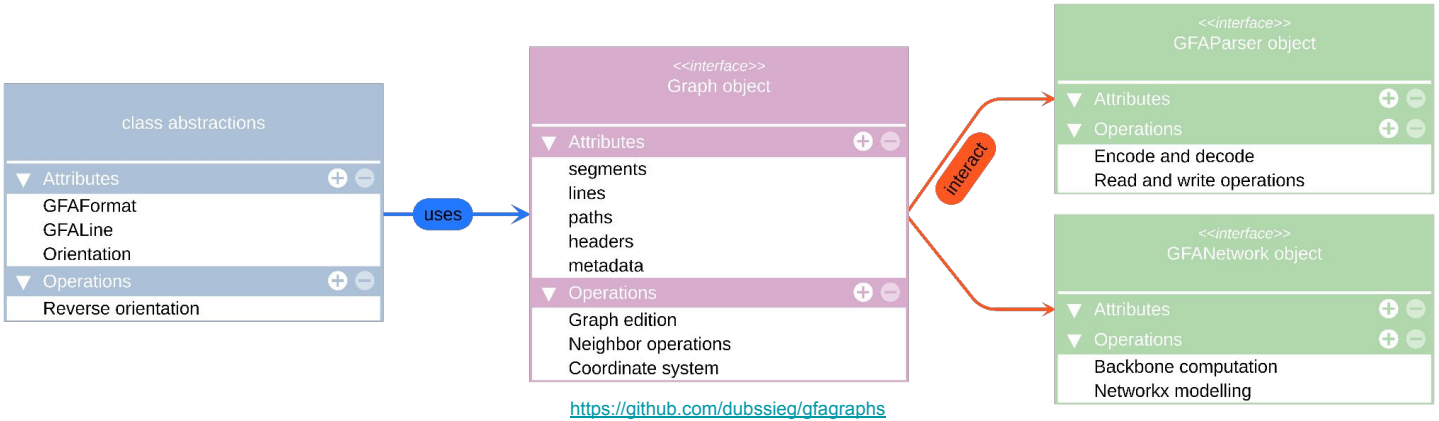
Library to handle GFA format:
- Interface to load, manipulate and query a pangenome graph

New visualisation software:
- Standalone interface with emphasis on haplotypes and variants
- Exploring differences between graphs made from the same genomes

Methods to interface graphs:
- Simplify graph with compression of substitution patterns
- Extract subgraphs in-between coordinates for visualisation



III. Library gfagraphs

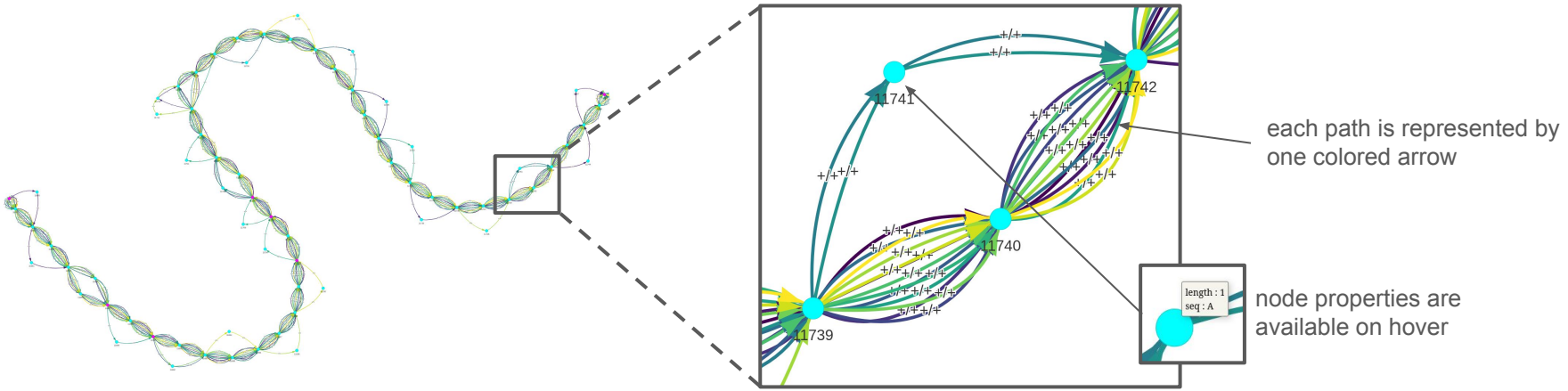


https://github.com/dubssieg/gfagraphs

Library to handle pangenome graphs

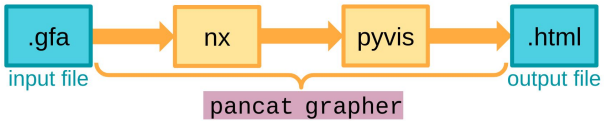
- Python library with documentation
- Various GFA formats (rGFA, GFA1, GFA1.1)
- Powers every tool presented afterwards
- Haplotype-based coordinates
- Neighbor system
- Edition operations

III.A. Displaying graphs with pancat grapher

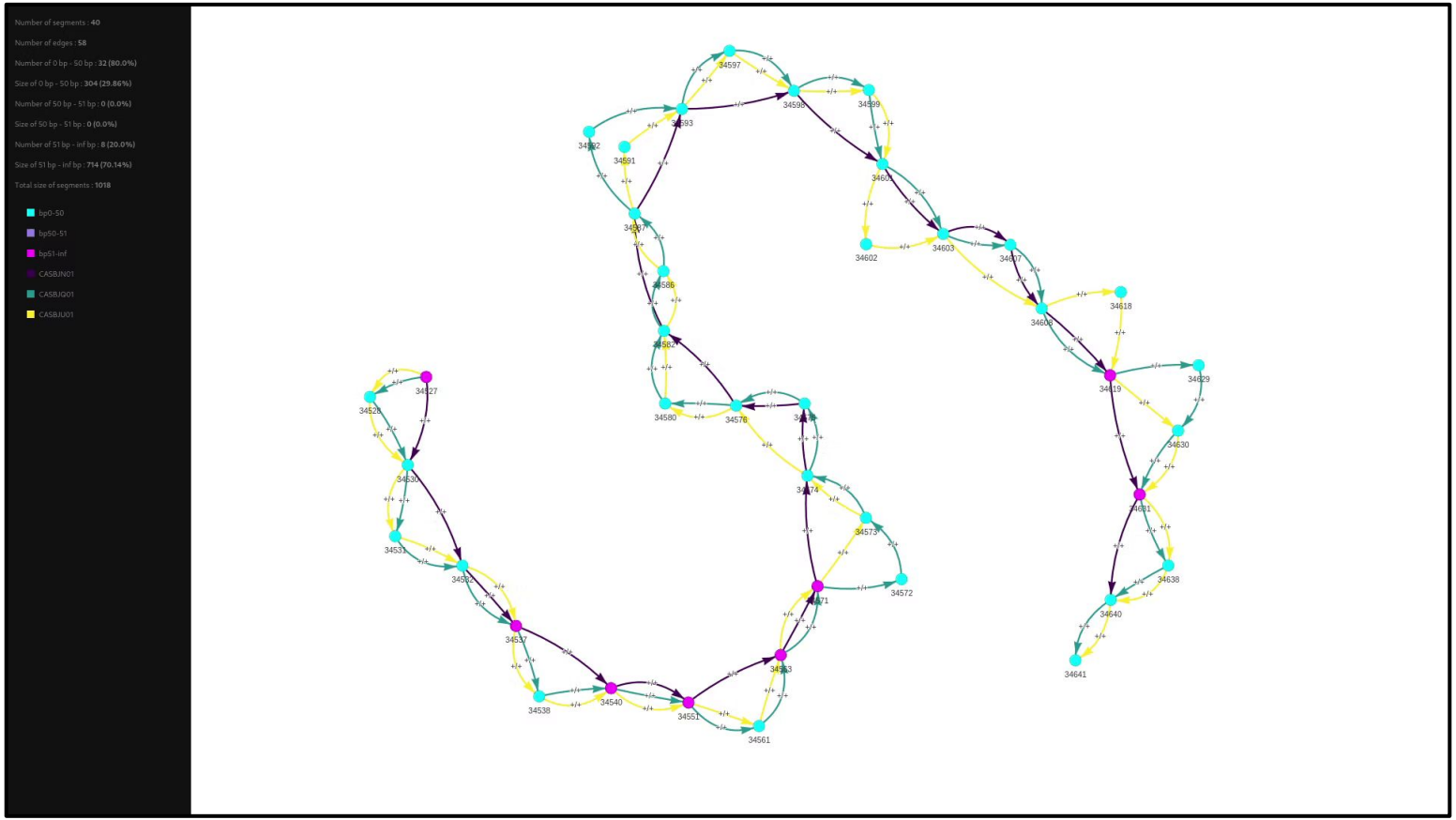


Pangenome viewer with a focus on variants

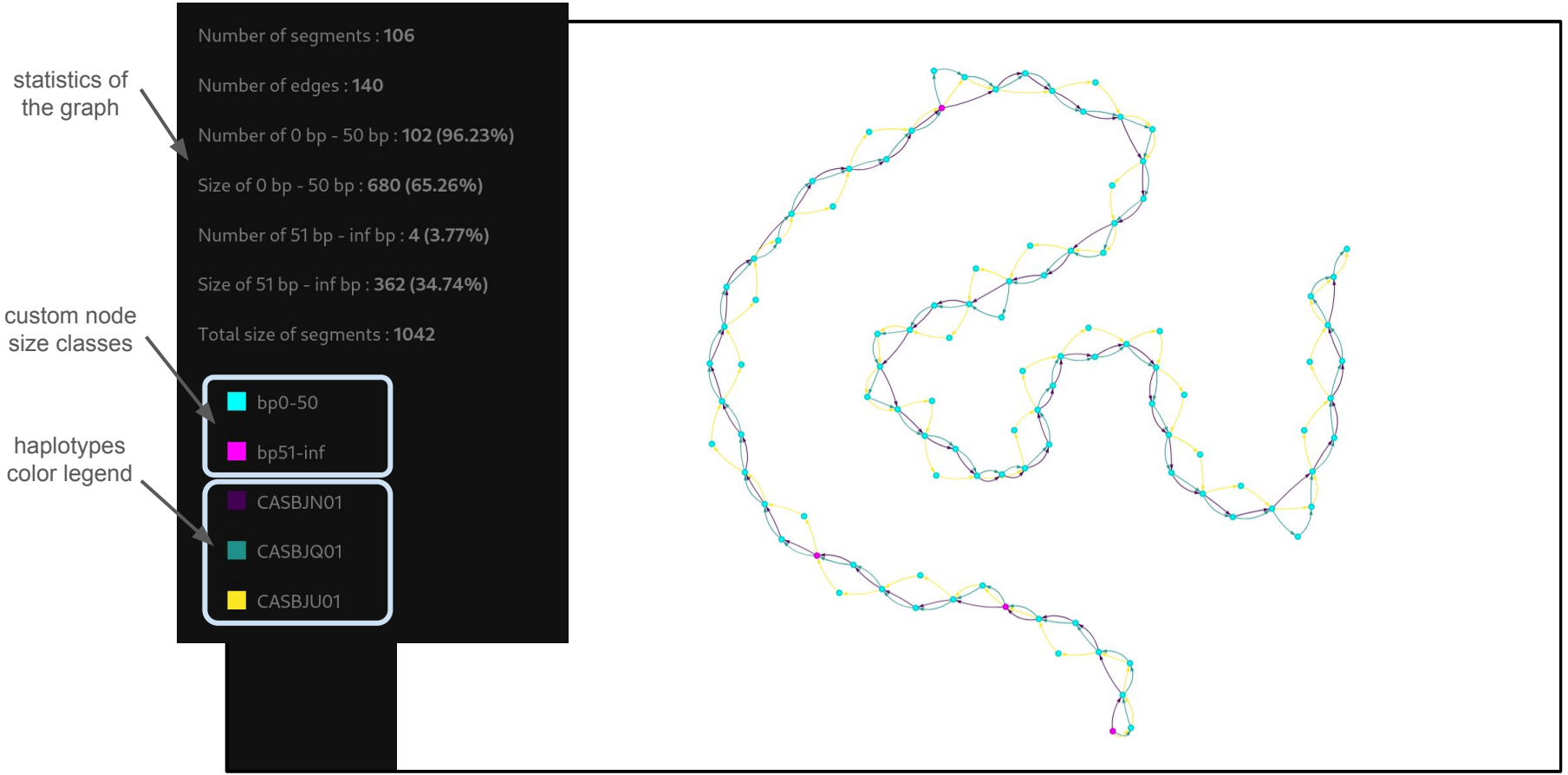
- Strong emphasis on the notion of haplotypes (paths)
- From a `.gfa` to a `.html` file displayable in any web browser
- Supports supplementary annotations of the `.gfa`



III.A. Displaying graphs with pancat grapher



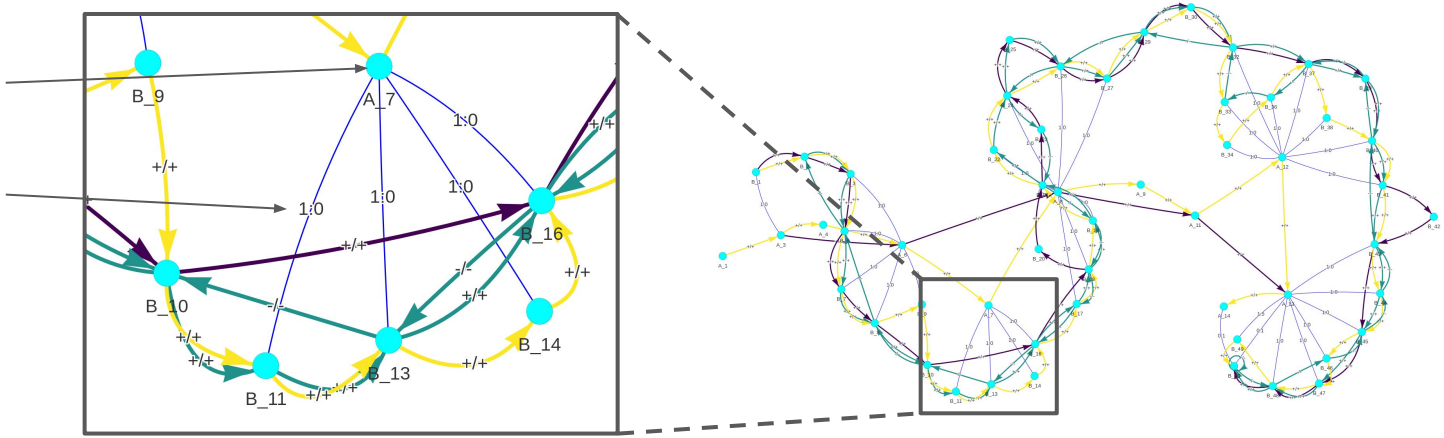
III.A. Displaying graphs with pancat grapher



III.B. Comparing with pancat multigrapher

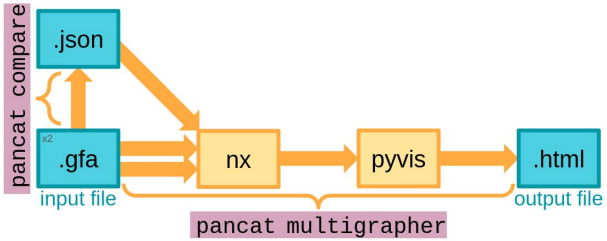
node 7 in graph A corresponds to nodes 11, 13, 14 and 16 in graph B

1:0 means 1 cut and 0 fusion to go from A7 to B11



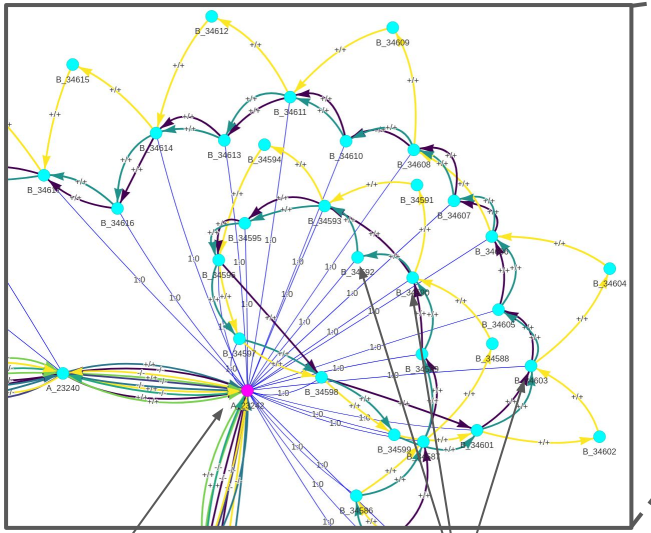
Side-to-side graph comparison visualisation

- Display the differences computed with `pancat compare`
- Graphs are linked by differences in nodes segmentation
- Visually align graphs by their shared genomes



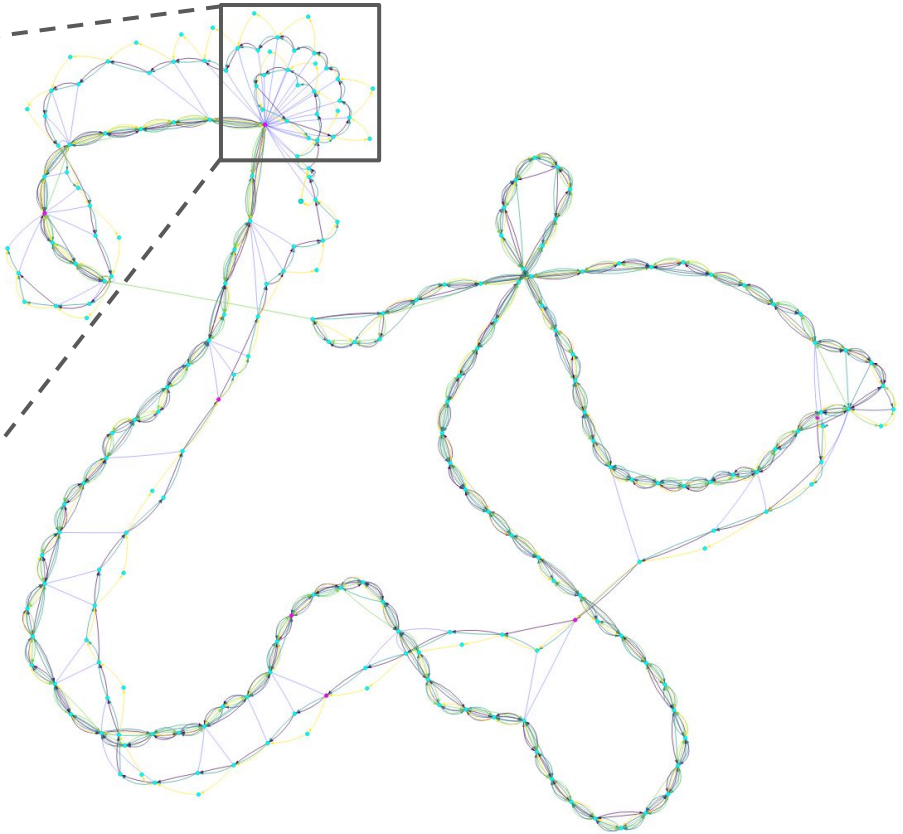
III.B. Comparing graphs

Hotspot of differences

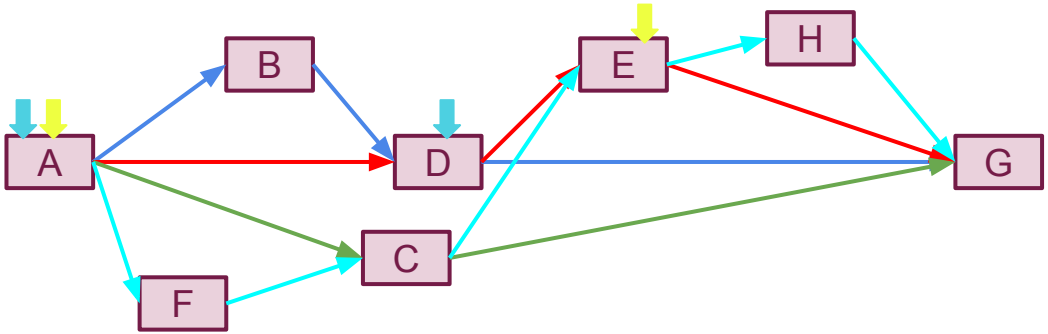


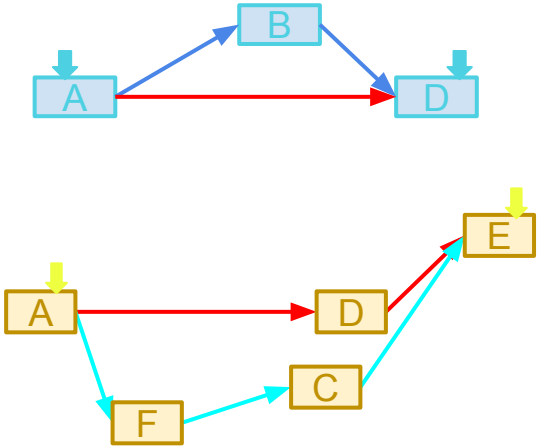
node of size 157 in PGGB graph

many small nodes in MGC graph



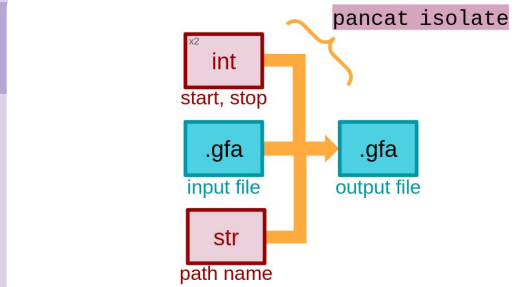
IV.A. Viewing and extracting subgraphs





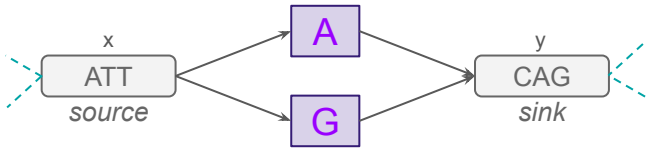
Select a zone of interest when performing a visualisation

- Intersection of paths that cross *source* and *sink*
- Subgraph between two path-based coordinates
- Integrated to the viewers and also standalone



IV.B. Reducing graphs with pancat compress

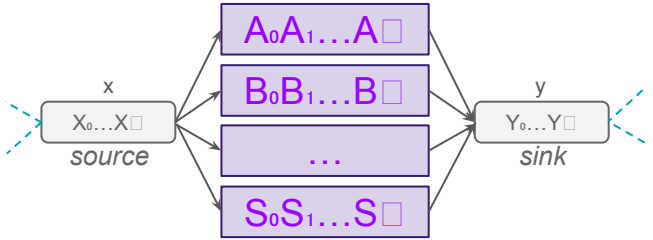
Simple case (1-sized substitution with two branches):



$\dot{W}$ = IUPAC(A,G) = **R**

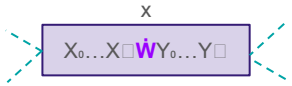
⇔

General case of a substitution of size up to *n* (with any number of branches):



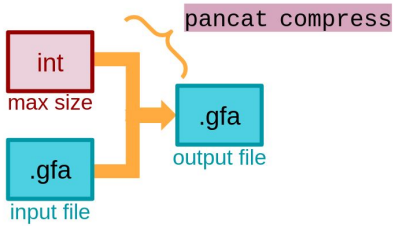
$\dot{W}$ = IUPAC(A_0,B_0,...,S_0) IUPAC(A_1,B_1,...,S_1) … IUPAC($A_□$,$B_□$,...,$S_□$)

⇔

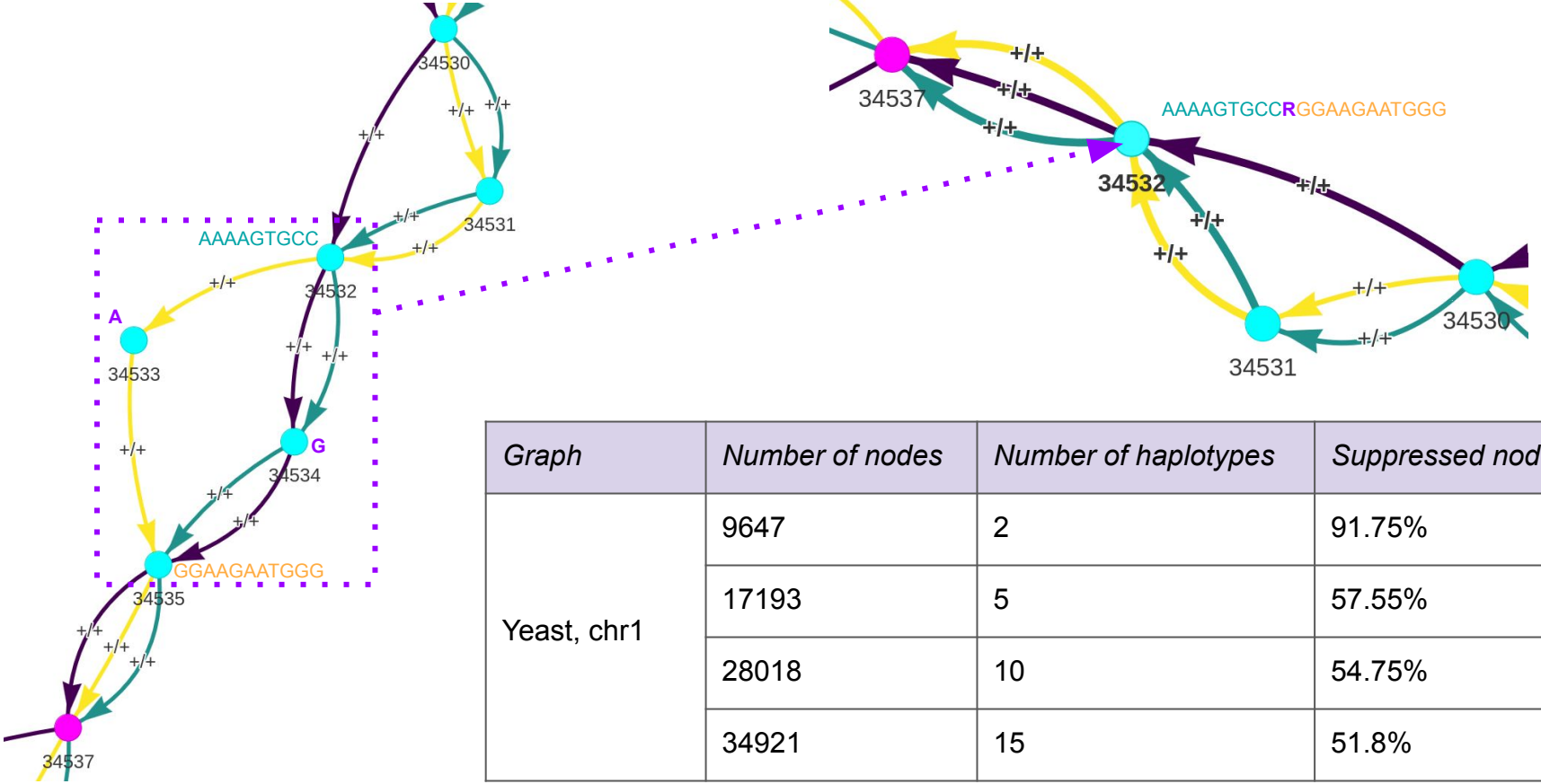


Reversible compression of pangenome graphs

- Compresses **substitutions** “without loss” (IUPAC code)
- Does not alter positions in graph, only topology
- Useful for graph traversal and visualisation



IV.B. Reducing graphs with pancat compress



Graph	*Number of nodes*	*Number of haplotypes*	*Suppressed nodes*
Yeast, chr1	9647	2	91.75%
	17193	5	57.55%
	28018	10	54.75%
	34921	15	51.8%

V. Scalability

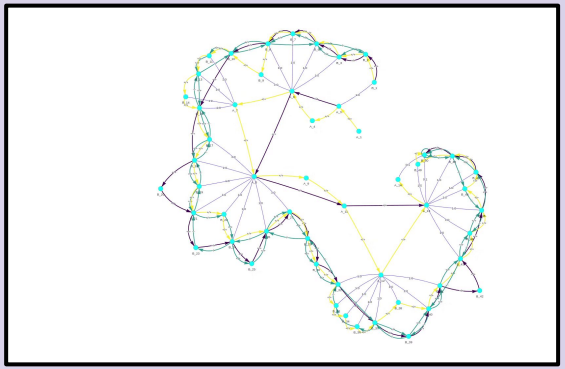
		Python		Rust	
Graph	*Size on disk*	*Memory*	*Loading time*	*Memory*	*Loading time*
Yeast *(chr 1, mgc)*	3.8MB	128.7MB	0.8s	5.4MB	0.3s
Yeast *(full, pggb)*	188.8MB	3.6GB	32s	1.3GB	6s
Human *(chr 21, mgc)*	982.8MB	12.0GB	95s	6.6GB	76s
Human *(chr 21, pggb)*	2.3GB	32.9GB	1092s	17.3GB	223s

Limitations linked to the python library

- High memory consumption
- Can scale to human chromosomes
- Time spent go up quickly (~30 min for chr1)
- Bottleneck is line parsing (95% of time spent)
- Perspective: backend migration (to rust)
- Half memory and time for similar object

VI. Discussion

Visualisation tool



Focus on path visualisation

Can display two graphs

Viewer as a standalone file

Interfacing methods

Substitution compression

Subgraph extraction

dubssieg/**pancat**

Pangenome graphs visualisation, distance computing, reconstruction of sequences and other utility functions

1 Contributor · 0 Issues · 31 Stars · 1 Fork

Perspectives

Scalability over whole pangenome graphs

Compression of other patterns than substitutions

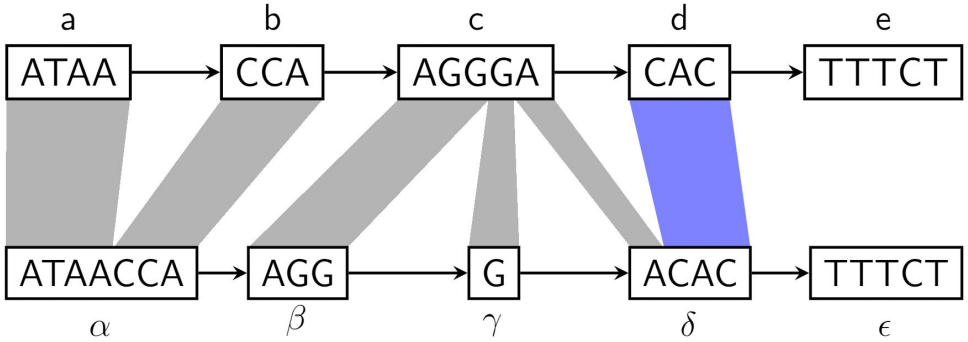
Definition of new levels of abstraction over graphs

Design tools from biological questions applications

https://github.com/dubssieg/pancat

Thanks for your attention!

Graph comparison

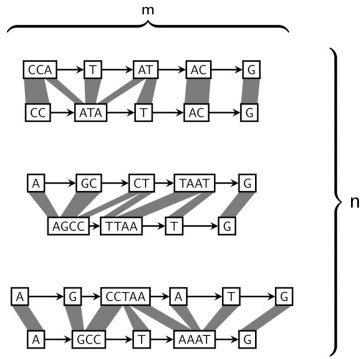


$$d_s(P_i^A, P_i^B) = min(|M| + |S|)$$

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

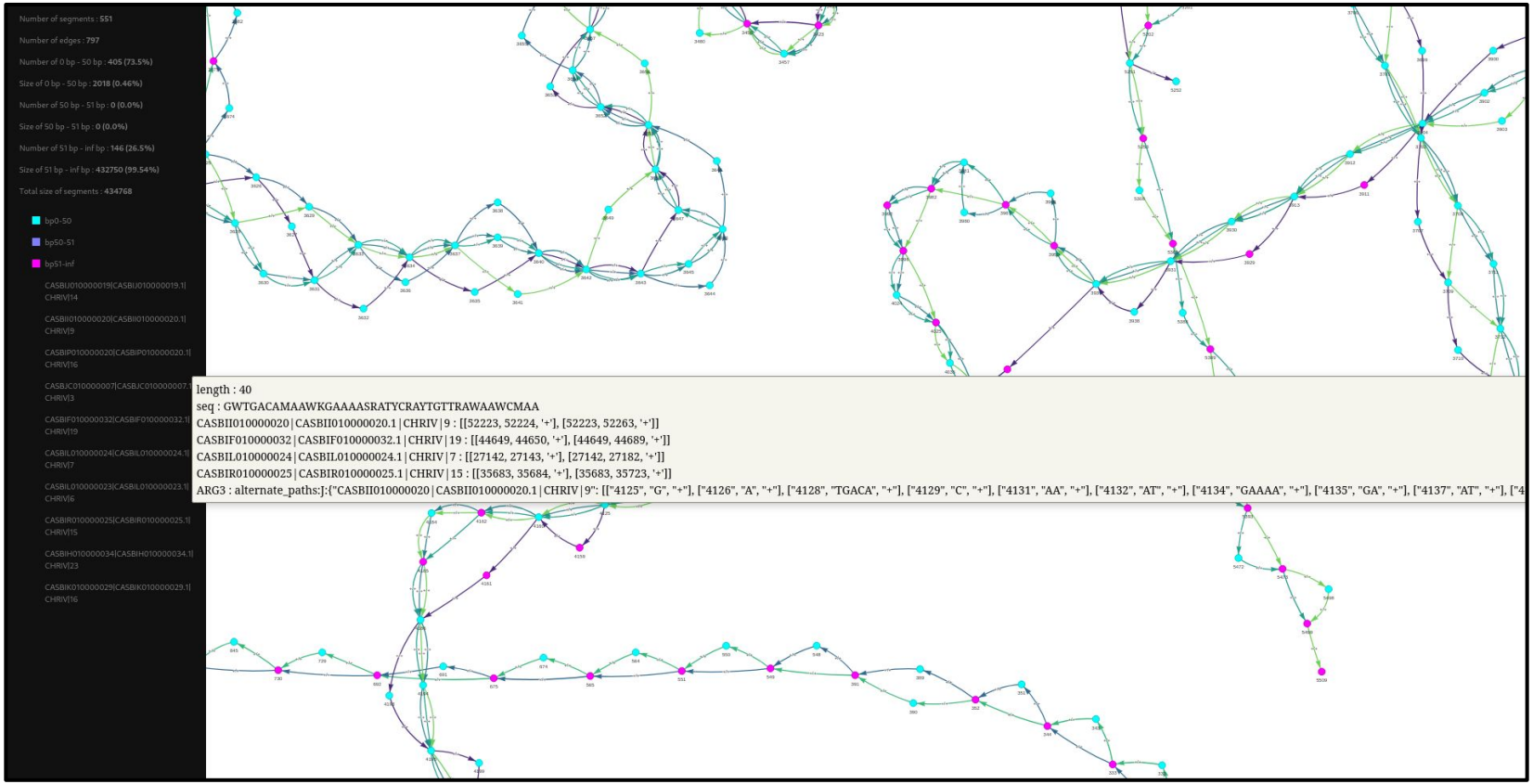
Distance established upon differences in segmentation

- Comparison of homologous paths
- A distance per path, with coordinates
- Two operations: *merge* and *split*
- Minimal count of operations between two paths



$O(n \times m)$

Supplementary tags



Some commands for pancat

Display 2kb of a simple graph with custom node size classes:

```
pancat grapher input_graph.gfa output_visu.html -b 2 10 50 1000 -s 23000 -e 25000 -r “refseq#path#0”
```

Display 2kb of two graphs displaying differences in-between them:

```
pancat multigrapher graph_A.gfa graph_B.gfa edits.json visu.html -s 23000 -e 25000 -r “refseq#path#0”
```

Reduce substitutions of size up to 50bp in a graph:

```
pancat compress input_graph.gfa -o output_graph.gfa -l 50
```

Extract a subgraph of 2kb along a selected genome:

```
pancat isolate input_graph.gfa output_graph.gfa -s 23000 -e 25000 -r “refseq#path#0”
```

Navigate the graph