Debugging long-read genome assemblies using string graph analysis

Pierre MARIJON, Jean Stéphane VARRÉ, Rayan CHIKHI

INRIA, Université Lille 1, CNRS

Why assemblies need debugging ?

Assembly of 3rd generation sequencing data

- requires correction (hybrid or non-hybrid)
- solves almost all genomic repetitions

Koren et Phillippy 2015 say "One chromosome, one contig", but ...

Why assemblies need debugging ?

Assembly of 3rd generation sequencing data

- requires correction (hybrid or non-hybrid)
- solves almost all genomic repetitions

Koren et Phillippy 2015 say "One chromosome, one contig", but ...

Bacterial assembly is not solved

NCTC : 3000 bacteria cultures sequenced with PacBio

521 out of 1136 assemblies are not single-contig

Species	Strain	Sample	Runs	Automated Assembly	Manual Assembly	Manual Assembly Chromosome Contig Number	Manual Assembly Plasmid Contig Number	Manual Assembly Unidentified Contig Number
Achromobacter xylosoxidans	NCTC10807	ERS451415	ERR550491 ERR550506 ERR550507	Pending	EMBL	1	0	0
Budvicia aquatica	NCTC12282	ERS462988	ERR581162	Pending	EMBL	2	0	0
Campylobacter jejuni	NCTC11351	ERS445056	ERR550473 ERR550476	Pending	EMBL	1	0	0
Cedecea neteri	NCTC12120	ERS462978	ERR581152 ERR581168 ERR597265	Pending	EMBL	7	1	0
Citrobacter amalonaticus	NCTC10805	ERS485850	ERR601566 ERR601575	Pending	EMBL	1	2	0
Citrobacter freundii	NCTC9750	ERS485849	ERR601559 ERR601565	Pending	EMBL	1	0	0
Citrobacter koseri	NCTC10849	ERS473430	ERR581173	Pending	EMBL	1	1	0
Corynebacterium diphtheriae	NCTC11397	ERS451417	ERR550510	Pending	EMBL	1	0	0
Cronobacter sakazakii	NCTC11467	ERS462977	ERR581151 ERR581167	Pending	EMBL	4	3	0
Enterobacter aerogenes	NCTC10006	ERS462975	ERR581148 ERR581149	Pending	EMBL	1	0	0
Enterobacter amnigenus	NCTC12124	ERS485854	ERR601570	Pending	EMBL	1	0	0
Enterobacter asburiae	NCTC12123	ERS485853	ERR601569 ERR601574	Pending	EMBL	2	3	0
Enterobacter cancerogenus	NCTC12126	ERS462979	ERR581153 ERR581169 ERR597266	Pending	EMBL	6	1	0

Bacterial assembly is not solved

NCTC : 3000 bacteria cultures sequenced with PacBio

521 out of 1136 assemblies are not single-contig

Species	Strain	Sample	Runs	Automated Assembly	Manual Assembly	Manual Assembly Chromosome Contig Number	Manual Assembly Plasmid Contig Number	Manual Assembly Unidentified Contig Number
Achromobacter xylosoxidans	NCTC10807	ERS451415	ERR550491 ERR550506 ERR550507	Pending	EMBL	1	0	0
Budvicia aquatica	NCTC12282	ERS462988	ERR581162	Pending	EMBL	2	0	0
Campylobacter jejuni	NCTC11351	ERS445056	ERR550473 ERR550476	Pending	EMBL	1	0	0
Cedecea neteri	NCTC12120	ERS462978	ERR581152 ERR581168 ERR597265	Pending	EMBL	7	1	0
Citrobacter amalonaticus	NCTC10805	ERS485850	ERR601566 ERR601575	Pending	EMBL	1	2	0
Citrobacter freundii	NCTC9750	ERS485849	ERR601559 ERR601565	Pending	EMBL	1	0	0
Citrobacter koseri	NCTC10849	ERS473430	ERR581173	Pending	EMBL	1	1	0
Corynebacterium diphtheriae	NCTC11397	ERS451417	ERR550510	Pending	EMBL	1	0	0
Cronobacter sakazakii	NCTC11467	ERS462977	ERR581151 ERR581167	Pending	EMBL	4	3	0
Enterobacter aerogenes	NCTC10006	ERS462975	ERR581148 ERR581149	Pending	EMBL	1	0	0
Enterobacter amnigenus	NCTC12124	ERS485854	ERR601570	Pending	EMBL	1	0	0
Enterobacter asburiae	NCTC12123	ERS485853	ERR601569 ERR601574	Pending	EMBL	2	3	0
Enterobacter cancerogenus	NCTC12126	ERS462979	ERR581153 ERR581169 ERR597266	Pending	EMBL	6	1	0

Why ?

Towards metagenomics

- Few datasets
- Lack of tailored assembler
- Will current genomic assemblers be adequate ?

Premise

An assembly graph can be defined as :

- nodes $\rightarrow$ reads
- edges $\rightarrow$ overlaps
- paths $\rightarrow$ contigs

Premise

An assembly graph can be defined as :

- nodes $\rightarrow$ reads
- edges $\rightarrow$ overlaps
- paths $\rightarrow$ contigs

We observe that :

- majority of assembly choice are made during graph construction
- hybrid or non-hybrid assemblers perform equally well
- $\rightarrow$ we will consider non-hybrid assembly

Assembly Graph

Best Overlap Graph

A graph with drastic selection of overlaps.

For each read we select two best overlaps : 1 left, 1 right.

BOGs are used by assemblers Canu [1] and HINGE [2].

1. Koren, Walenz et al. 2017.
2. Kamath et al. 2017.

Full Overlap Graph

A graph with maximal information.

For each node we keep all overlaps.

FOGs are generated by Minimap PAF output, used by Miniasm [3].

3. Li 2016.

Dataset used

- One bacterial dataset :
  - **Terriglobus roseus :** synthetic, 20x coverage (LongISLND [4])
- One metagenomic dataset :
  - **MBRAC-5 :** synthetic, 5 bacterias from [5]

4. Lau et al. 2016.
5. Singer et al. 2016.

Debugging tools

How to debug assemblies ?

Two datasets that do not assemble well :

Dataset	Number of Canu contig	Number of Miniasm contig	Expected
Terriglobus roseus	3	7	1
MBRAC-5	18	85	5

3 assembly graphs : FOG, Canu BOG, Miniasm's graph.

How to debug assemblies ?

Two datasets that do not assemble well :

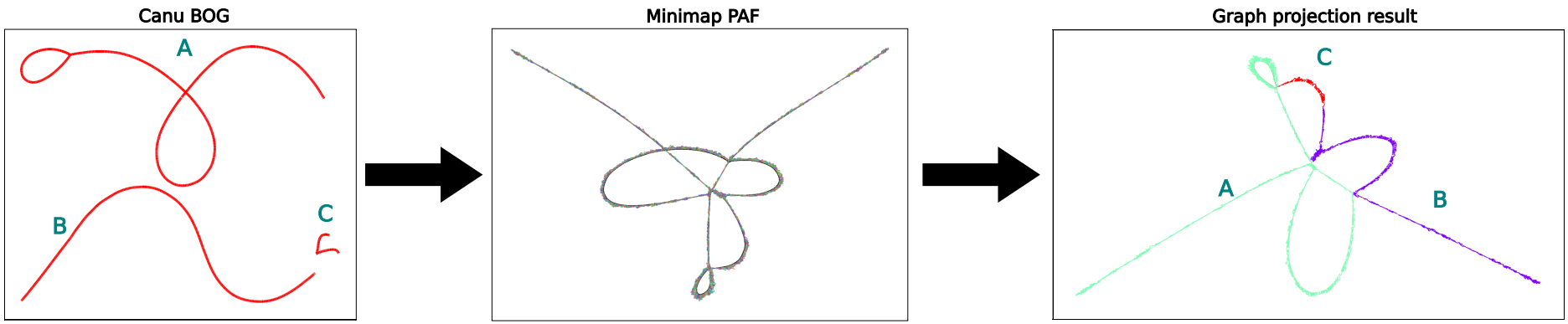
Dataset	Number of Canu contig	Number of Miniasm contig	Expected
Terriglobus roseus	3	7	1
MBRAC-5	18	85	5

3 assembly graphs : FOG, Canu BOG, Miniasm's graph.

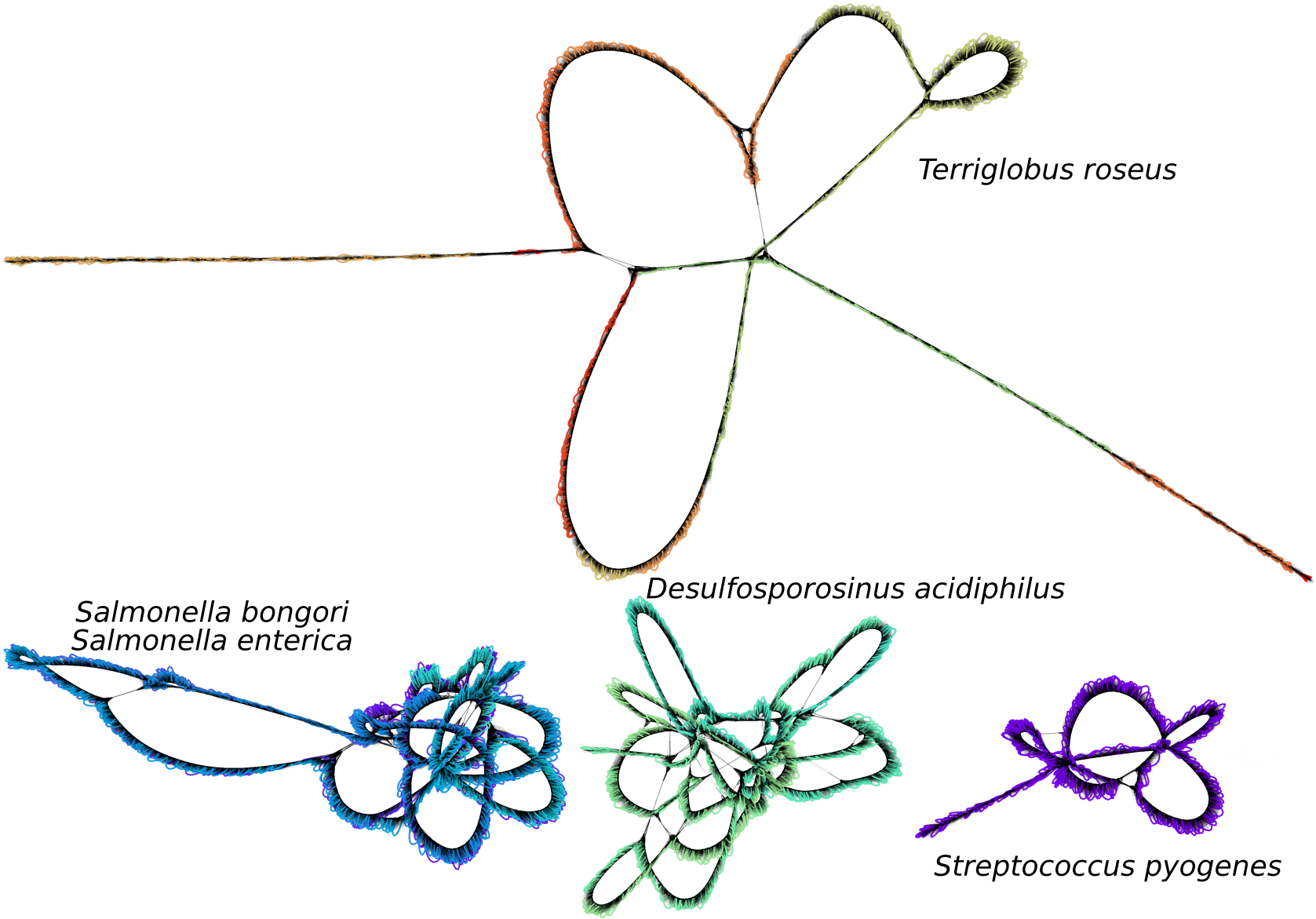
We will compare the assembly graphs.

Graph projection

Graph projection : of a selective graph (BOG) onto a less selective graph (FOG)

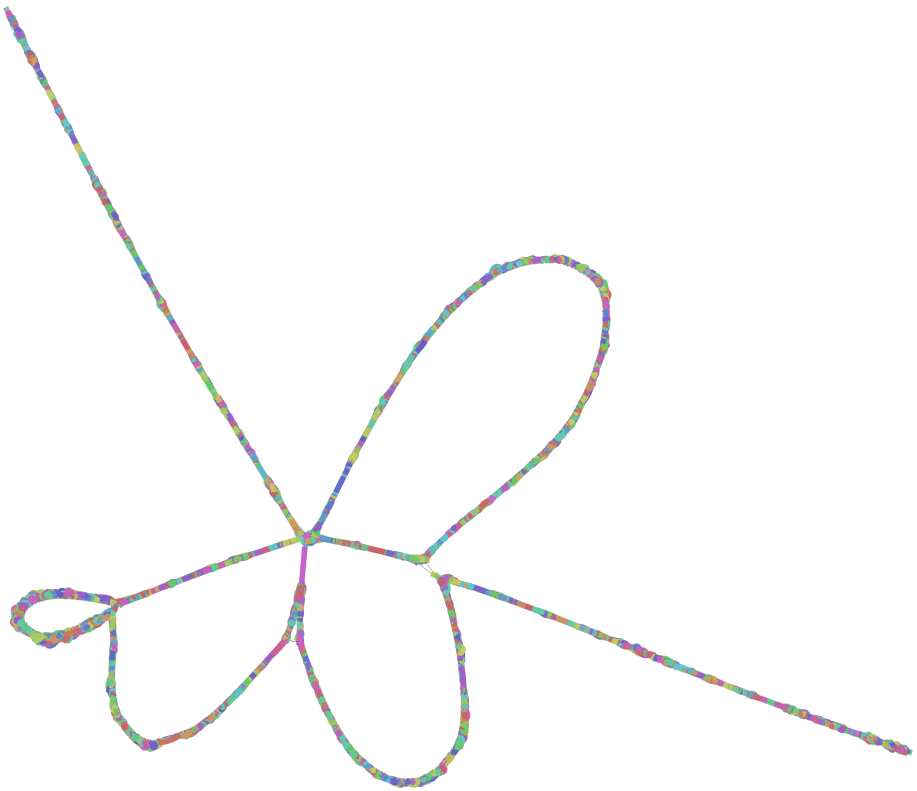


Metagenomics graph projection (annotated)

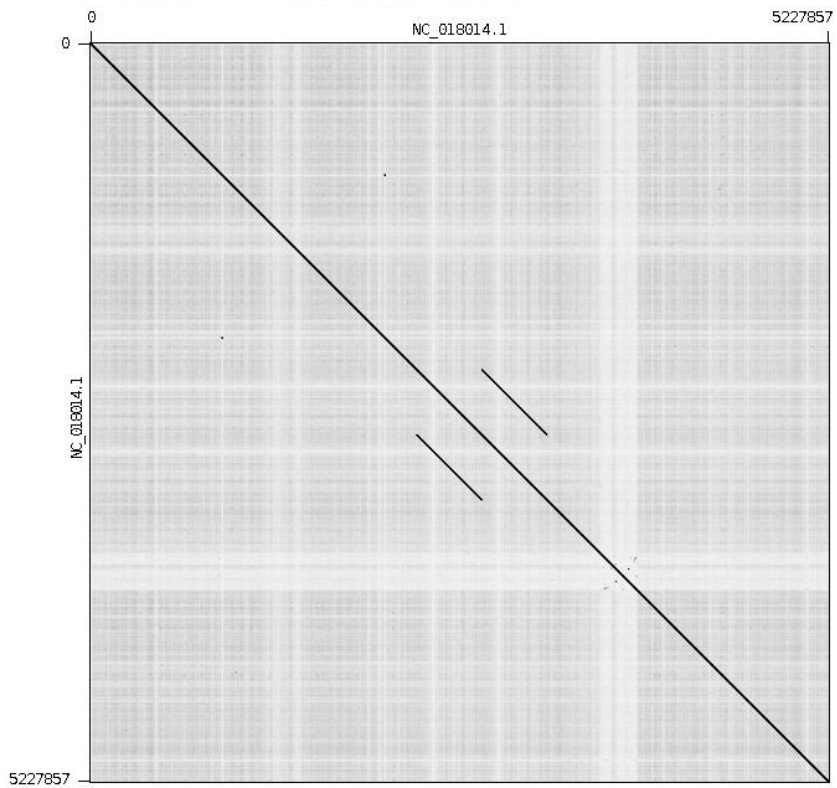


MBRAC-5 Canu BOG on Minimap FOG

Full Overlap Graph of one bacteria

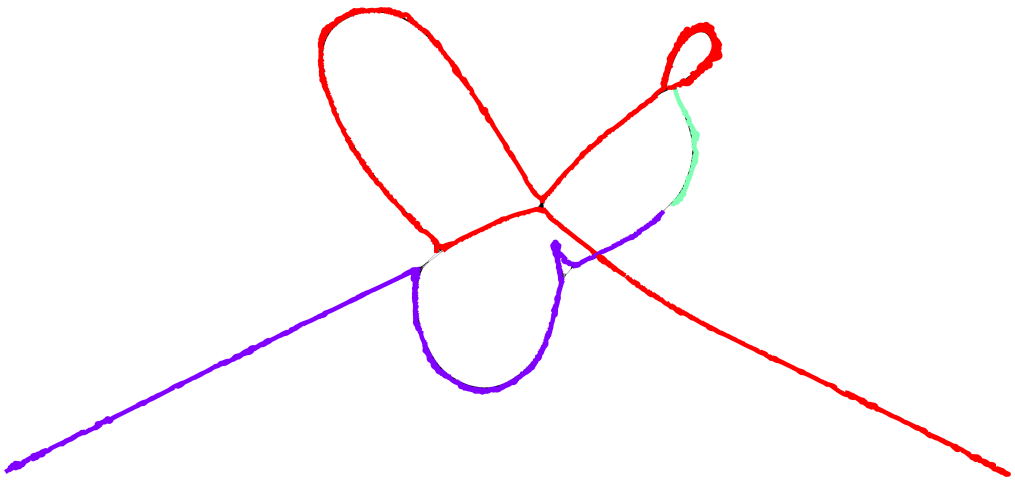


Minimap FOG graph of **Terriglobus roseus**

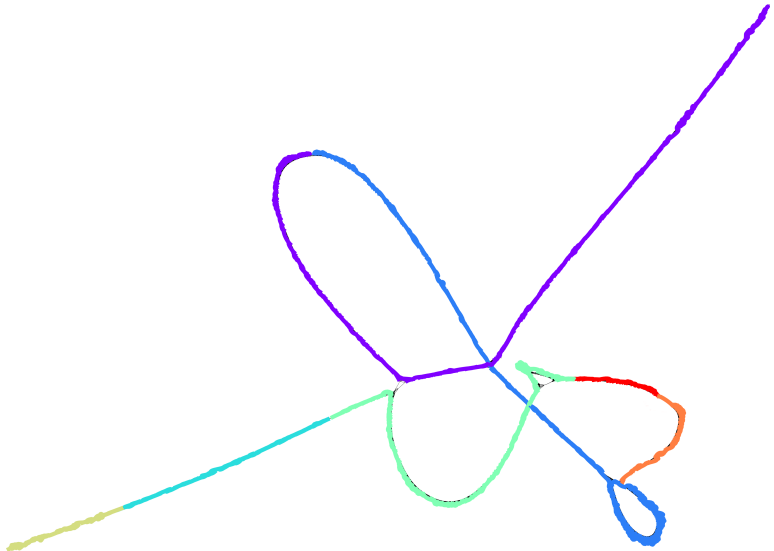


dotplot *T. roseus*, genome vs genome

Comparing projections across assembler

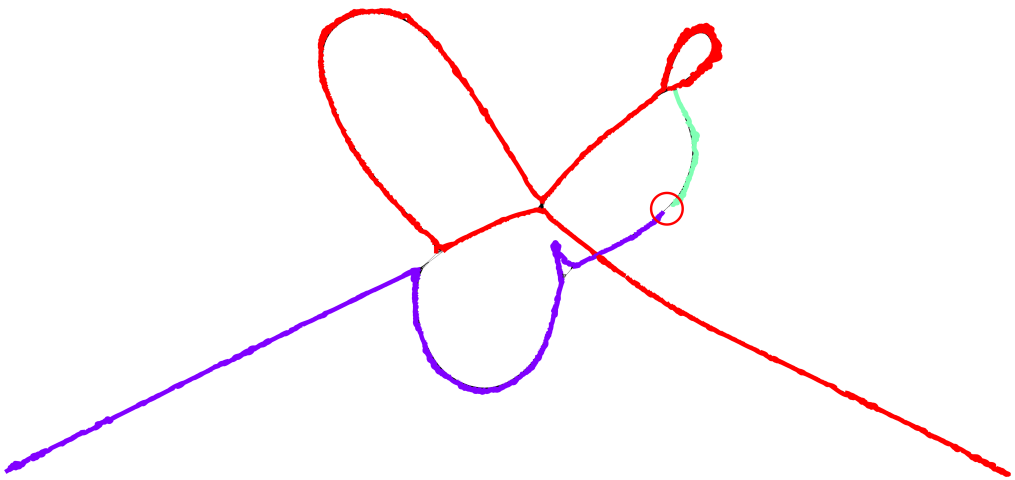


Canu BOG project on Minimap FOG

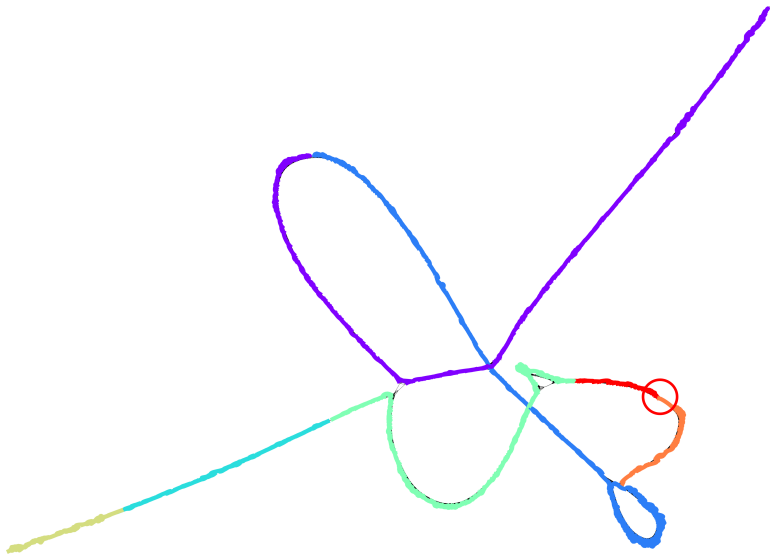


Miniasm assembly graph on FOG

Comparing projections across assembler

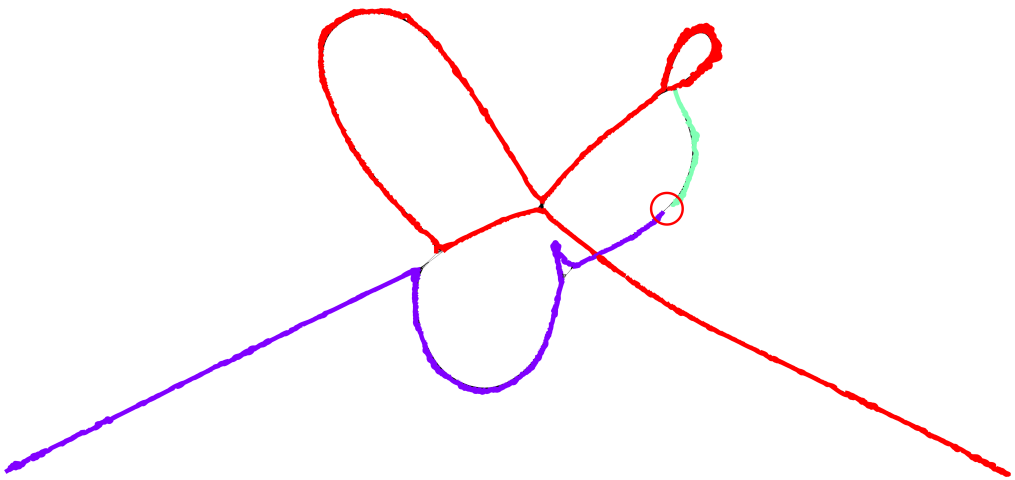


Canu BOG project on Minimap FOG

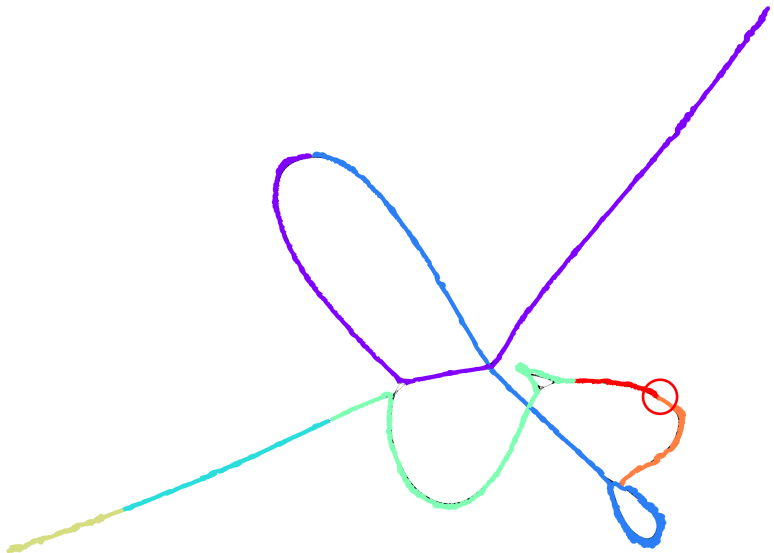


Miniasm assembly graph on FOG

Comparing projections across assembler



Canu BOG project on Minimap FOG

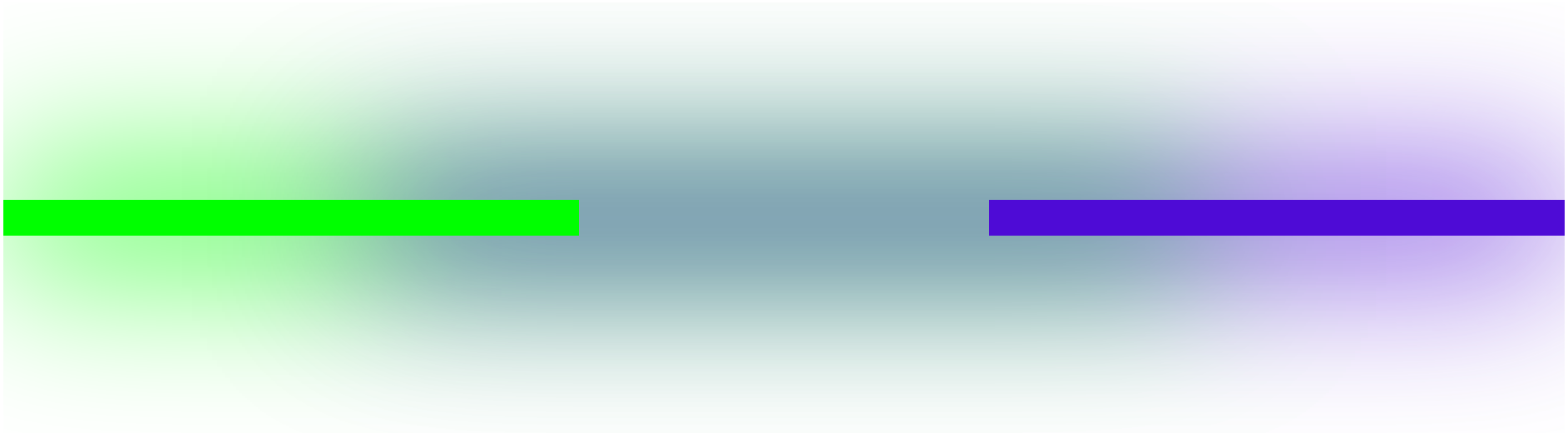


Miniasm assembly graph on FOG

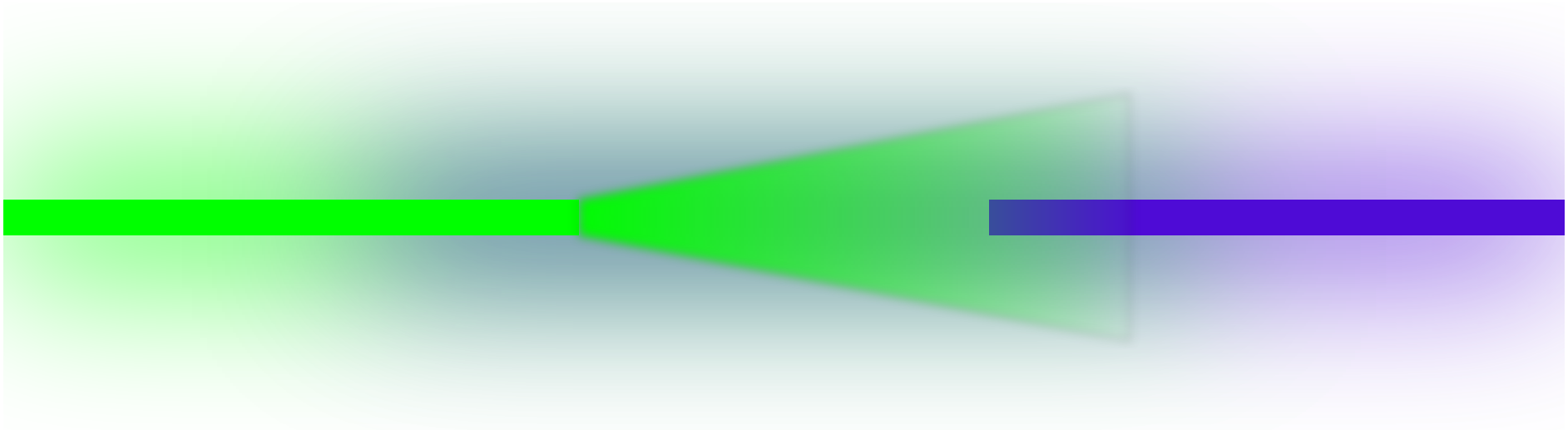
This *assembly breakpoint* cannot be :

- explained by a repetition,
- nor solved by assembly reconciliation

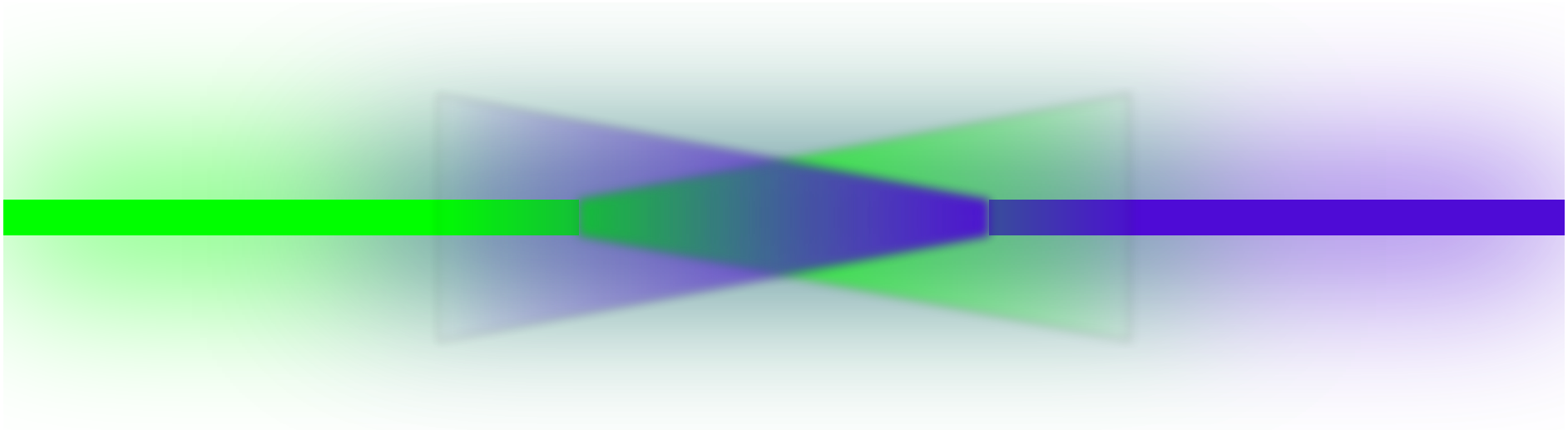
Subgraph extraction



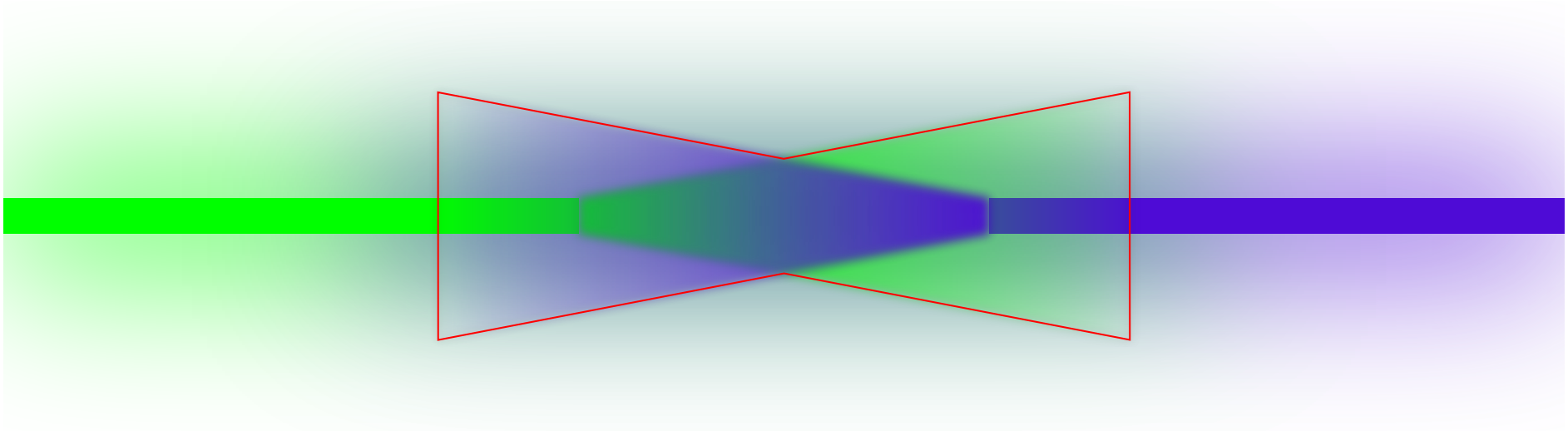
Subgraph extraction



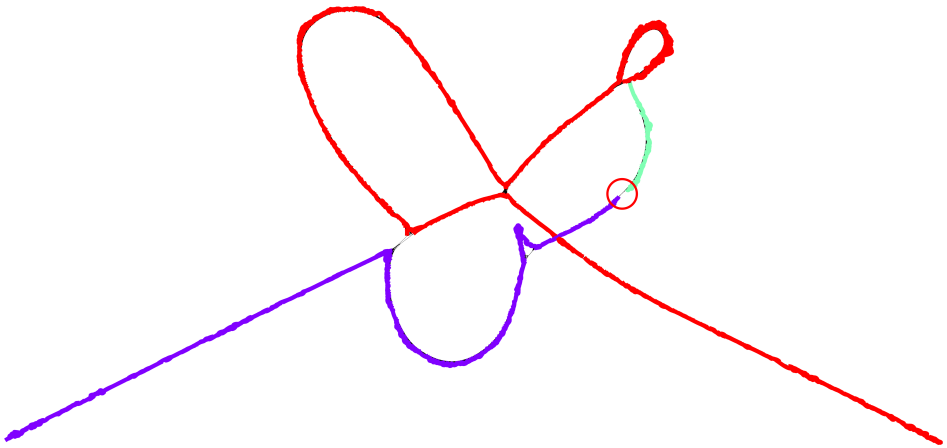
Subgraph extraction

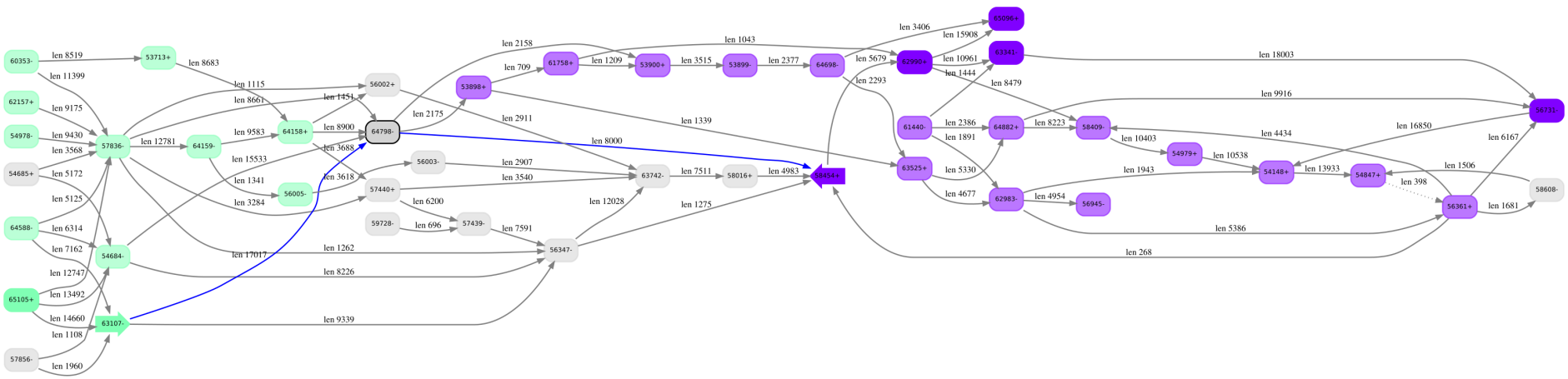


Subgraph extraction



Subgraph extraction





Conclusion

- Bacterial assembly is not solved
- Study of assembly graphs can help
- Graph projection pin-points where assemblies break
- Subgraph extraction enables to understand why

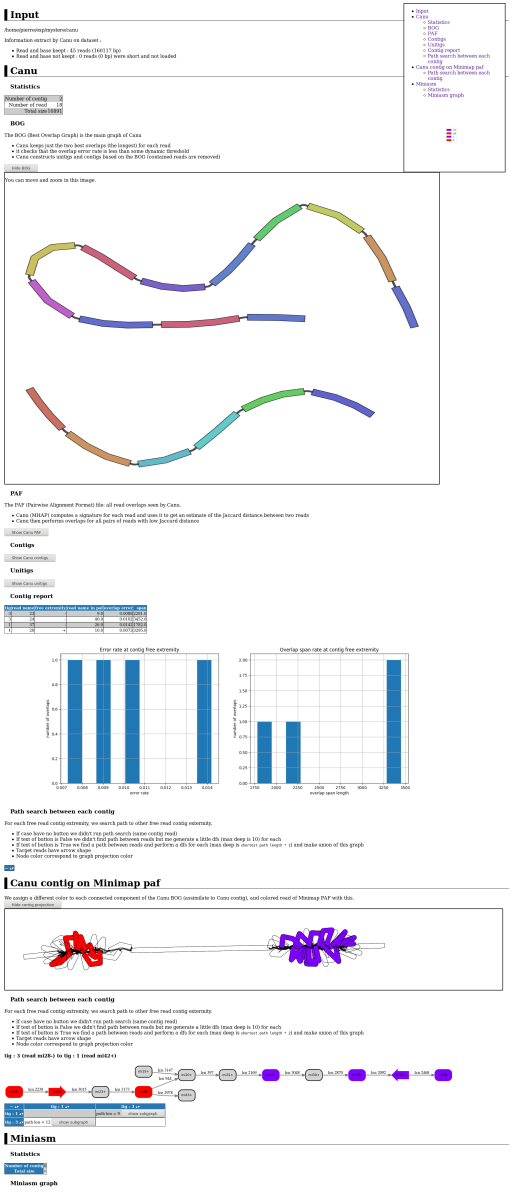
If your 3rd generation assembly needs debugging..

We created a pipeline to run our analysis easily with a fancy HTML output.

`https://gitlab.inria.fr/pmarijon/assembly_report`

Contacts :

- mail : pierre.marijon@inria.fr
- twitter : @pierre_marijon

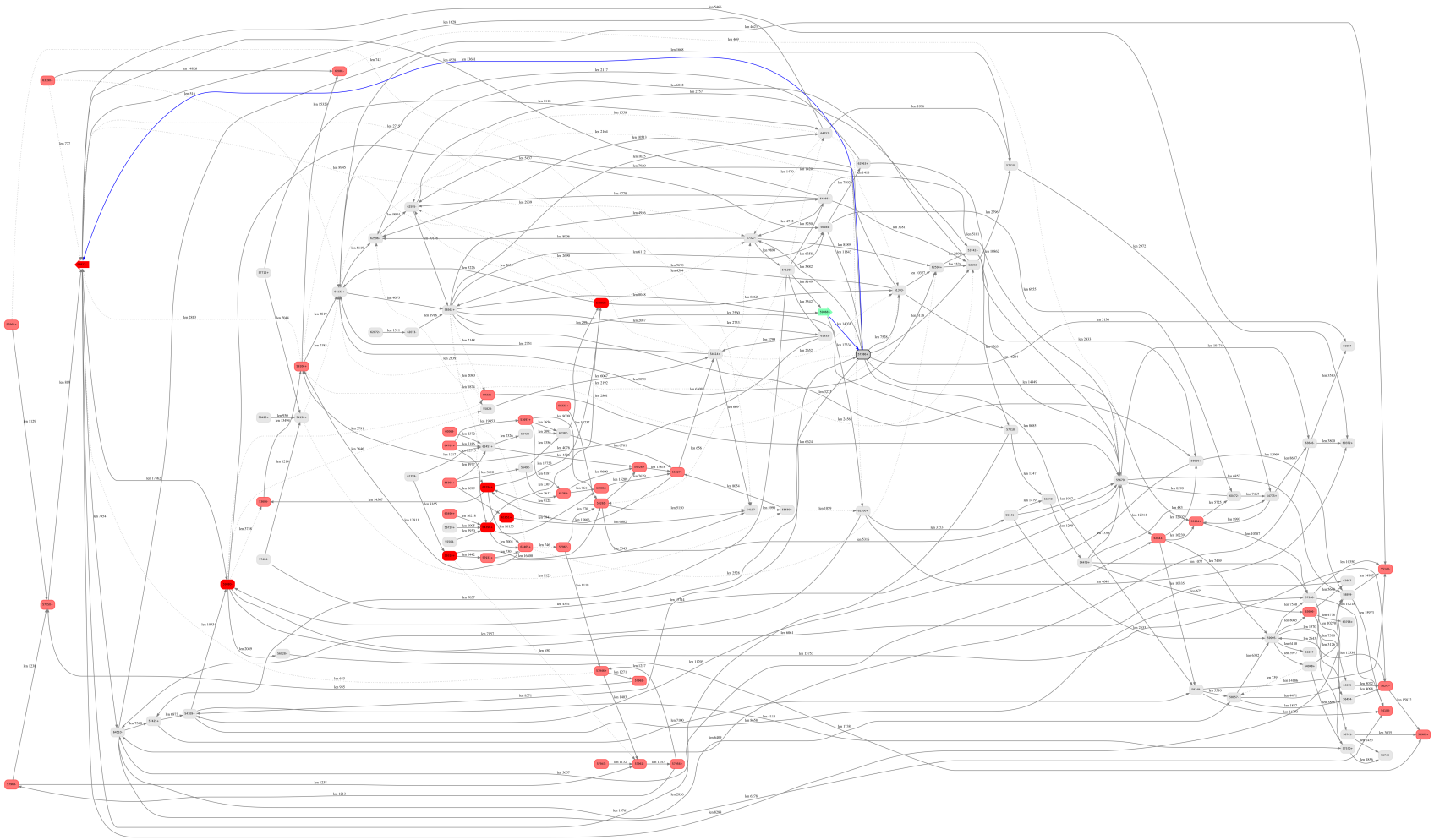


Future

- Find better layout for subgraph visualization
- NCTC dataset analysis (or your dataset ?)
- How to visualize a large FOG

Future

- Find better layout for subgraph visualization
- NCTC dataset analysis (or your dataset ?)
- How to visualize a large FOG



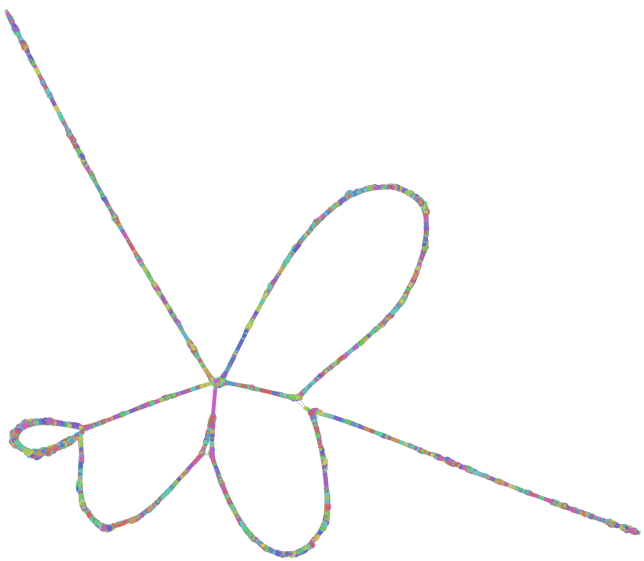
Future

- Find better layout for subgraph visualization
- NCTC dataset analysis (or your dataset ?)
- How to visualize a large FOG

SRA id	NCTC number of contig	Canu number of contig
ERS530422	6	7
ERS523588	7	10
ERS513137	7	12
ERS530437	6	13
ERS530440	7	8
ERS485853	5	13
ERS530413	6	7
ERS718603	5	9
ERS538530	6	7
ERS715425	6	10

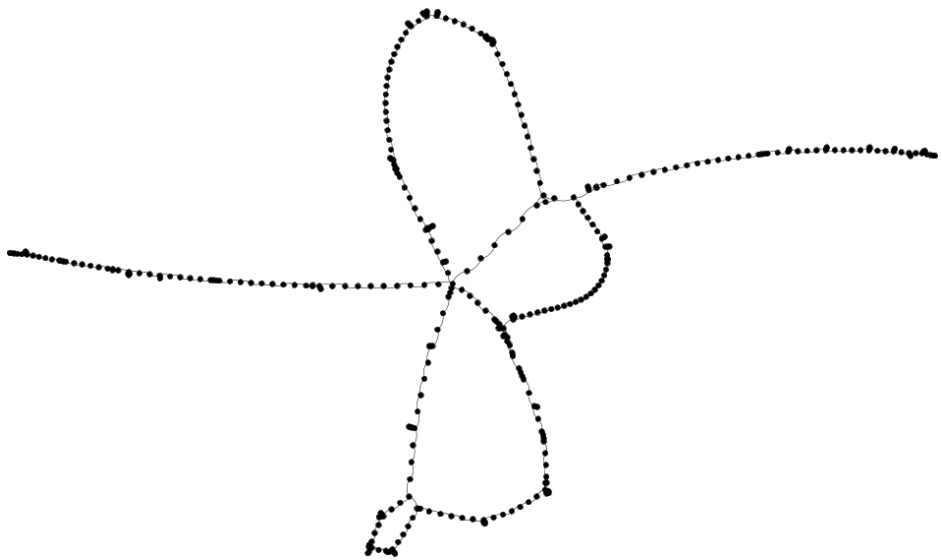
Future

- Find better layout for subgraph visualization
- NCTC dataset analysis (or your dataset ?)
- How to visualize a large FOG



Terriglobulus Roseus PAF :

11,381 nodes, 122,153 edges



Terriglobulus Roseus Compressed PAF :

368 nodes, 400 edges ; MATAM algorithm [Pericard *et al* 2017]

If your 3rd generation assembly needs debugging..

We created a pipeline to run our analysis easily with a fancy HTML output.

`https://gitlab.inria.fr/pmarijon/assembly_report`

Contacts :

- mail : pierre.marijon@inria.fr
- twitter : @pierre_marijon