Assembly improvement:
based on Ragout approach

student: Anna Lioznova
scientific advisor: Son Pham

Plan

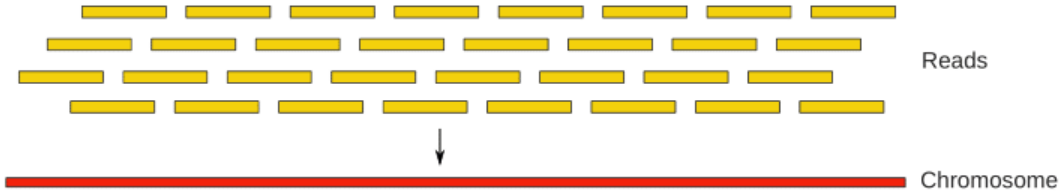
- Ragout overview
  - Datasets
- Assembly improvements
  - Quality
    - overlap graph
    - paired-end reads
  - Coverage

Plan

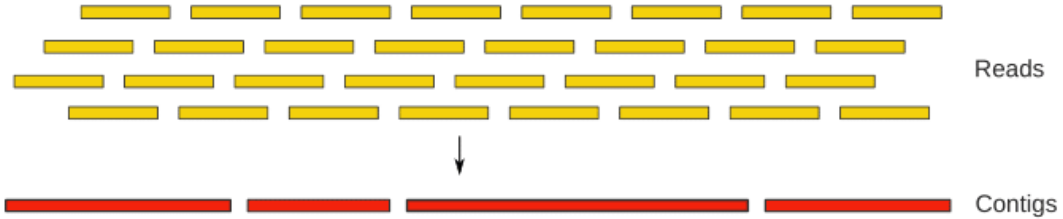
- **Ragout overview**
  - Datasets
- Assembly improvements
  - Quality
    - overlap graph
    - paired-end reads
  - Coverage

Genome assembly

Expectation



Reality



Ragout

Reference-Assisted Genome Ordering UTility

https://github.com/fenderglass/Ragout

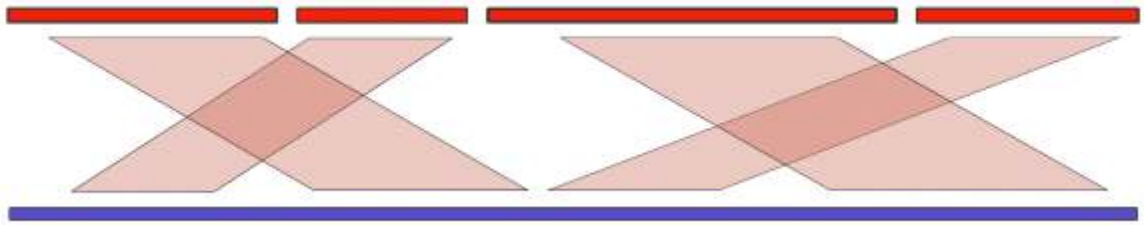
- Ingredients:
  - Multiple references
  - Contigs/scaffolds from short-read assembly
  - Phylogenetic tree
- Output: scaffolds

Reference-assisted assembly

Naive approach

Using a complete genome of another closely-related organism

Contigs are being aligned on that reference genome



Multiple references

The more references we use, the more information we have, the less is the number of misassemblies

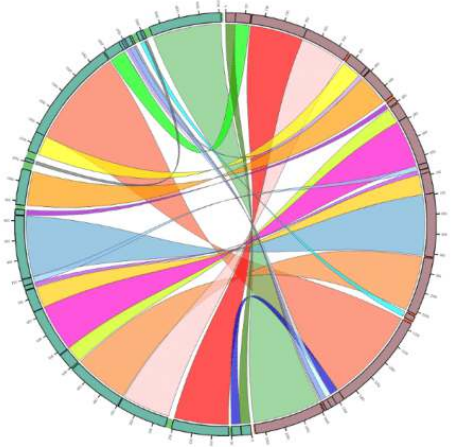
["Ragout — a reference-assisted assembly tool for bacterial genomes", Mikhail Kolmogorov et al., Vol. 30 ISMB 2014, pages i302–i309, doi:10.1093/bioinformatics/btu280]

Synteny blocks

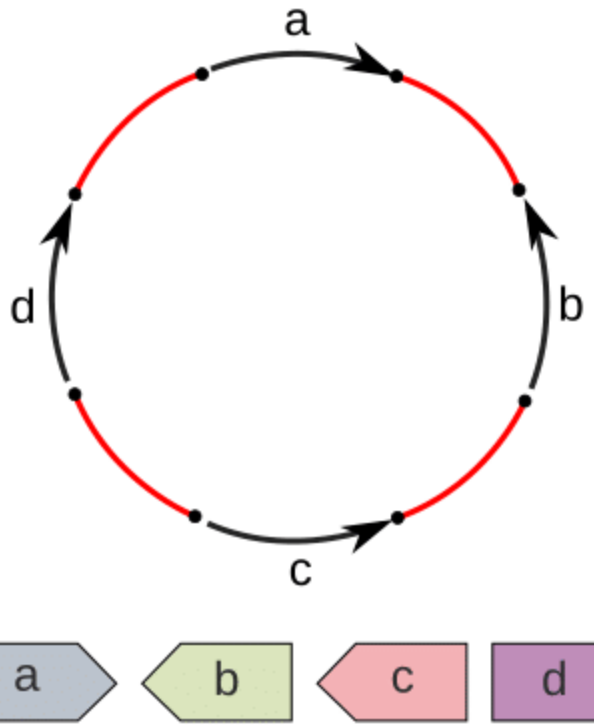
To compare multiple genomes we extract conserved segements – synteny blocks

- Align contigs to reference(s)
- Maf2synteny for different block size

https://github.com/fenderglass/maf2synteny



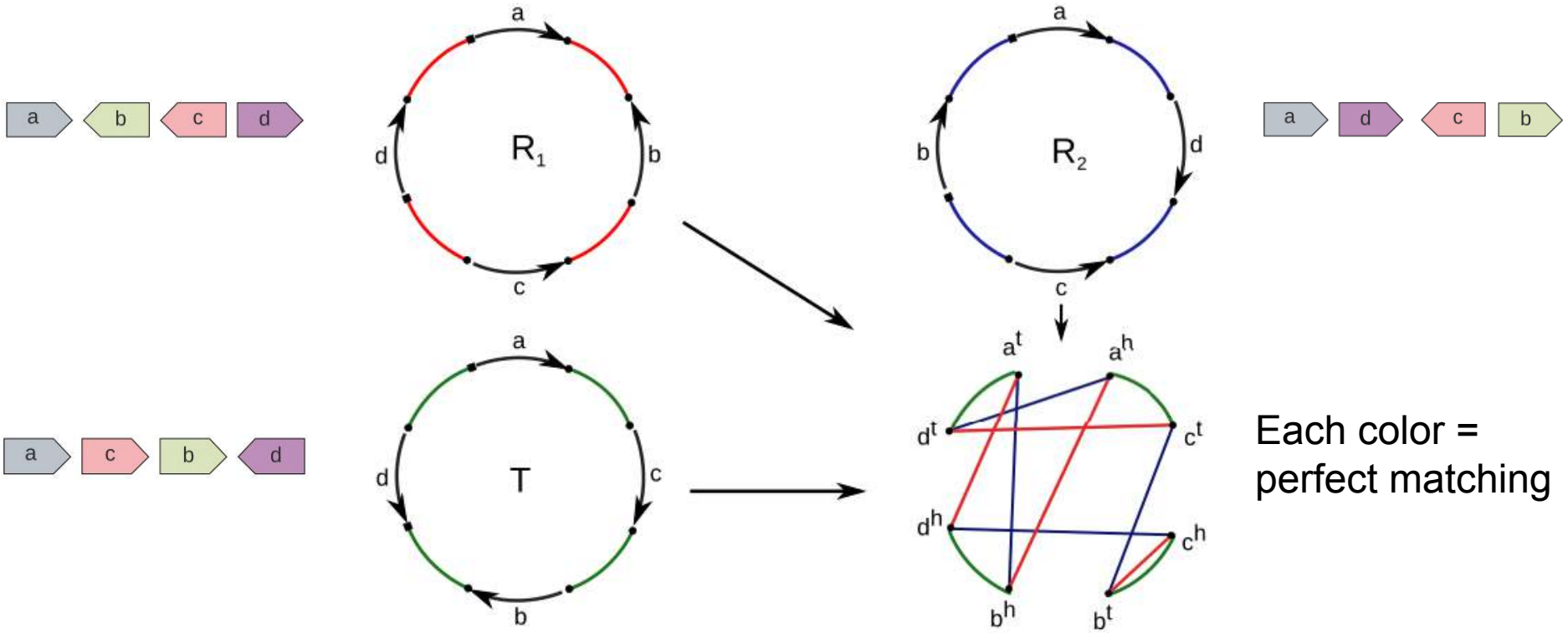
Synteny blocks and adjacencies



Black edges (directed) = synteny blocks

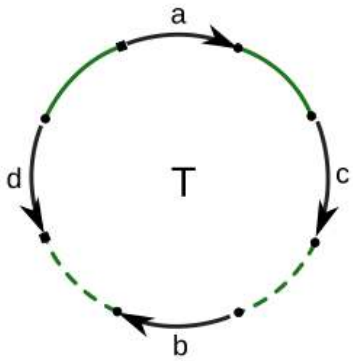
Red edges = adjacencies of synteny blocks

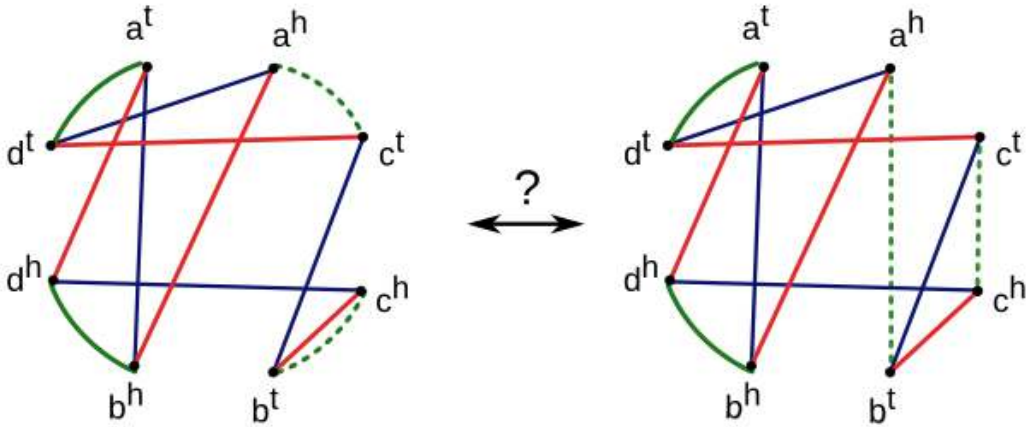
Breakpoint graphs

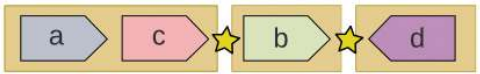


Each color = perfect matching

Incomplete breakpoint graph







Some adjacencies are missing

Find missing edges
= **Recover perfect matching**
Multiple variants of such matching
=> parsimony procedure

Ragout pipeline:

Algorithm 1 Ragout pseudocode

```
procedure RAGOUT(references, target, phylogeny, blockSizes)
    assemblies ← ∅
    for all blockSize in blockSizes do
        synBlocks ← RUNSIBELIA(references, target, blockSize)
        bpGraph ← BUILDBREAKPOINTGRAPH(synBlocks)
        weightedGraph ← EDGESSCORE(bpGraph, phylogeny)
        adjacencies ← MINPERFMATCHING(weightedGraph)
        scaffolds ← BUILDSCAFFOLDS(target, adjacencies)
        ADD(scaffolds, assemblies)
    end for
    scaffolds ← MERGEITERATIONS(assemlies)
    assemblyGraph ← BUILDASSEMBLYGRAPH(target)
    scaffolds ← REFINESCAFFOLDS(scaffolds, assemblyGraph)
    OUTPUTSCAFFOLDS(scaffolds)
end procedure
```

Plan

- Ragout overview
  - **Datasets**
- Assembly improvements
  - Quality
    - overlap graph
    - paired-end reads
  - Coverage

Bacterial datasets

- E.Coli
- H.Pylori
- S.Aureus
- V.Cholerae

Drosophila dataset

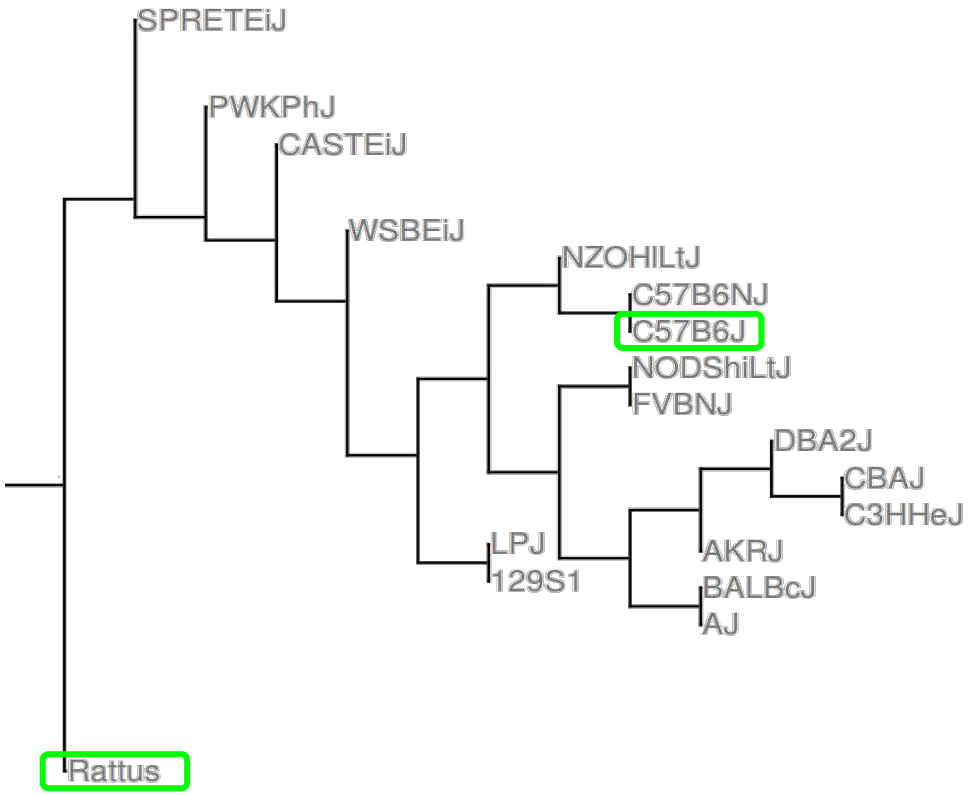
10^8 bp, 6 chromosomes

- References
  - miranda
  - yakuba
  - melanogaster
  - simulans
- Simulated reads (error free)
  - yakuba

Mice dataset

10^9 bp, 20 chromosome pairs

- 18 species
- 2 complete



Phylogenetic tree

Plan

- Ragout overview
  - Datasets
- Assembly improvements
  - Quality
    - **overlap graph**
    - paired-end reads
  - Coverage

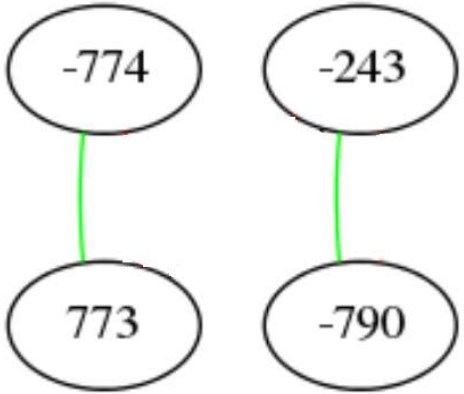
Assembly problem

a perfect matching in reference genome, but links are not present in the target genome

Example of the problem

Edges:

reference

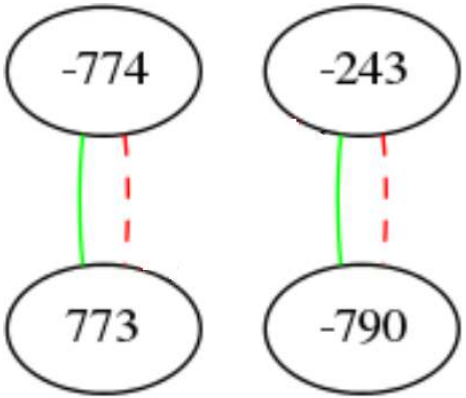


Example of the problem

Edges:

reference

predicted



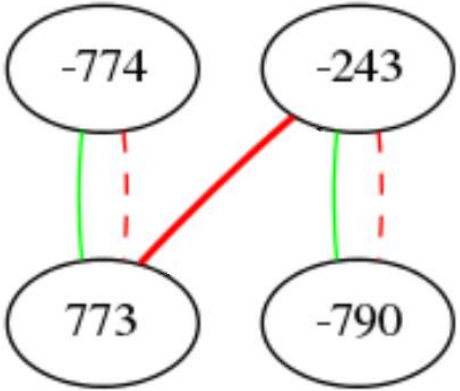
Example of the problem

Edges:

target

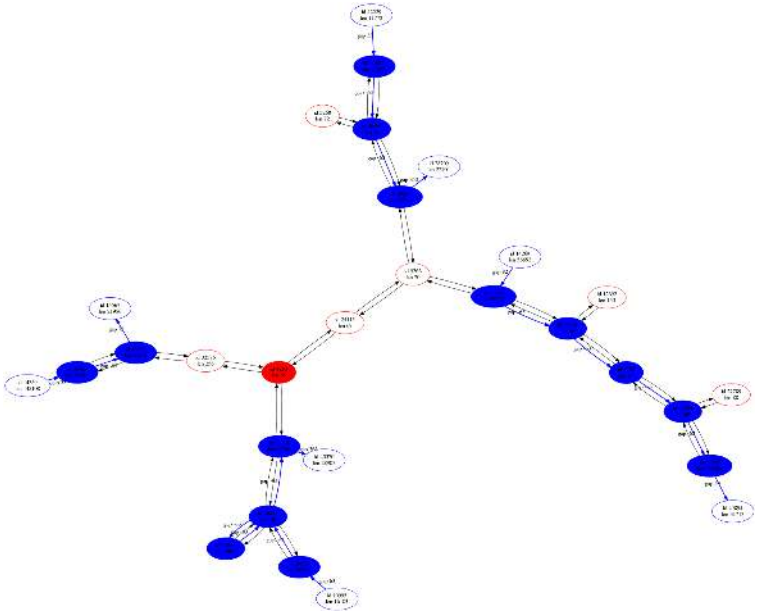
reference

predicted



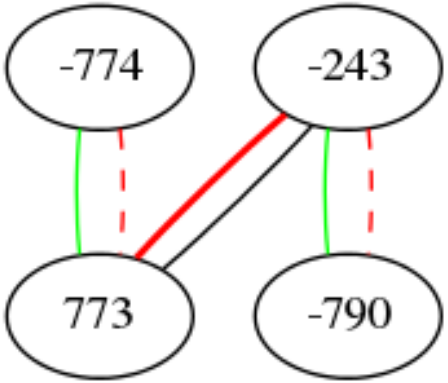
Links from contigs overlap graph

Add edges to break point graph from contigs overlap graph



Overlap graph links: expectation

New edges will resolve misassemblies, caused by edges in reference genomes



Overlap graph links: reality

Drosophila dataset

- 52 misassemblies
- 33 black edges
- 6 “good” black edges

6 correct edges will improve assembly, whilst
27 wrong edges will add misassemblies

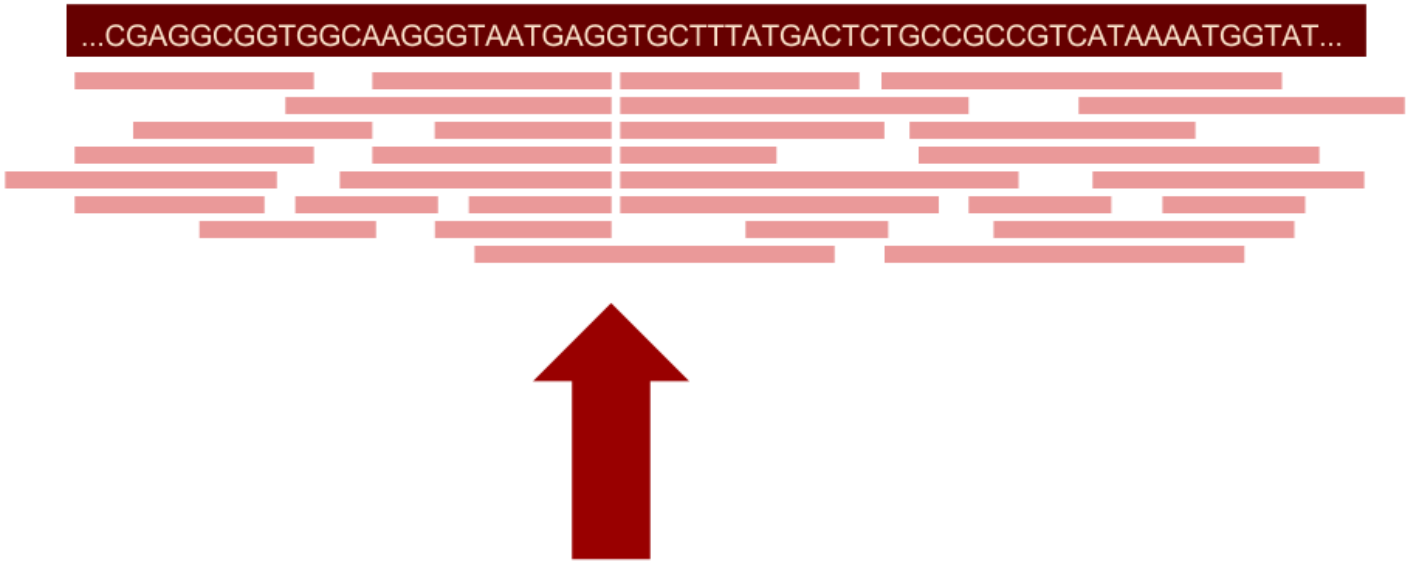
Plan

- Ragout overview
  - Datasets
- Assembly improvements
  - Quality
    - overlap graph
    - **paired-end reads**
  - Coverage

Paired-end reads: idea

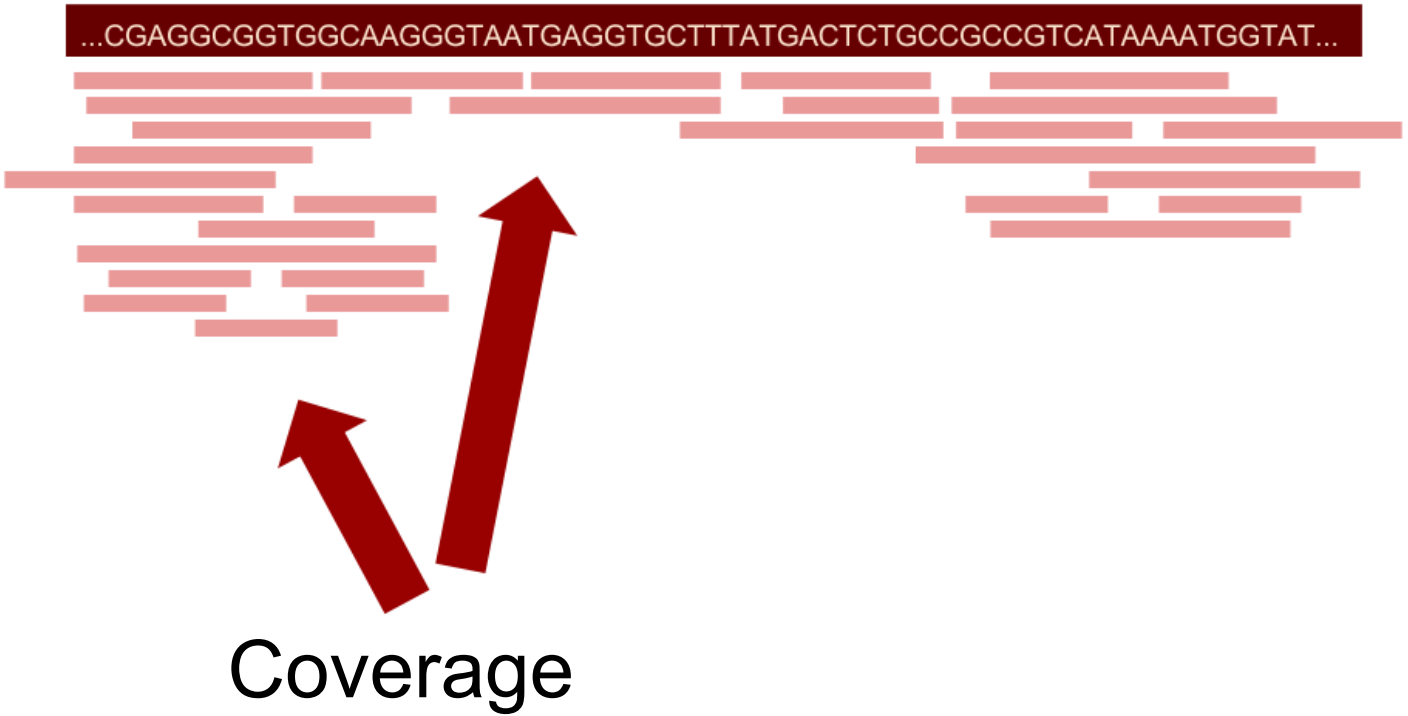
Align paired-end reads to scaffolds and detect misassemlies

Quality control: paired-end reads

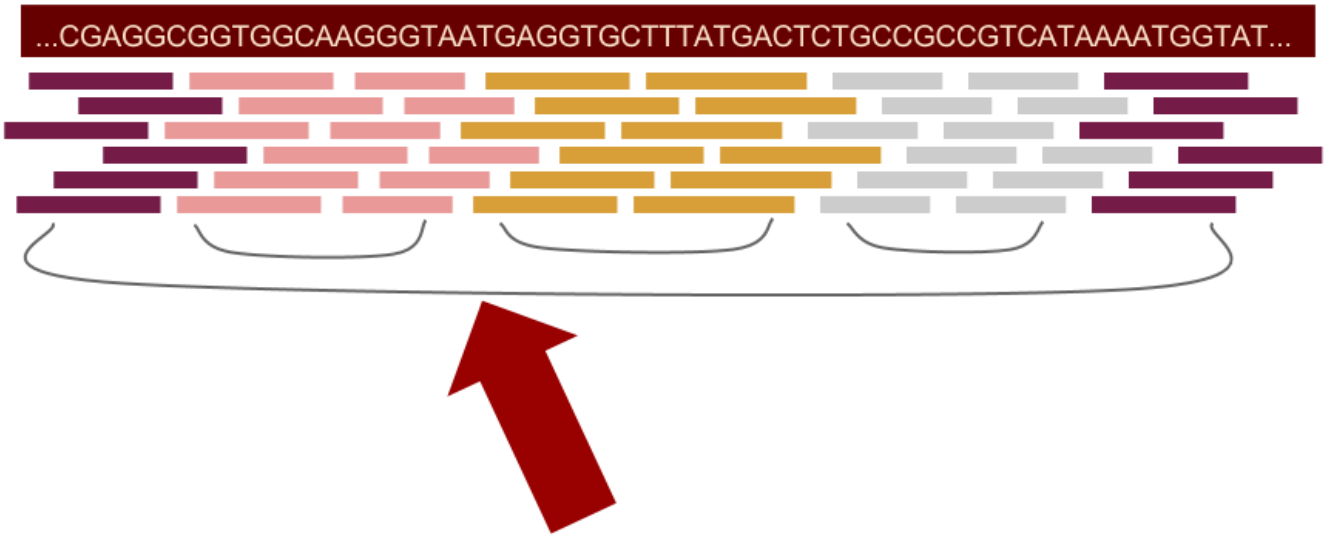


Ends and begins of reads

Quality control: paired-end reads



Quality control: paired-end reads



Insert size

Paired-end reads: reality

Alignment takes too long, we did it only once for the mice.

There are other metrics to estimate assembly quality: primers, transcripts, etc

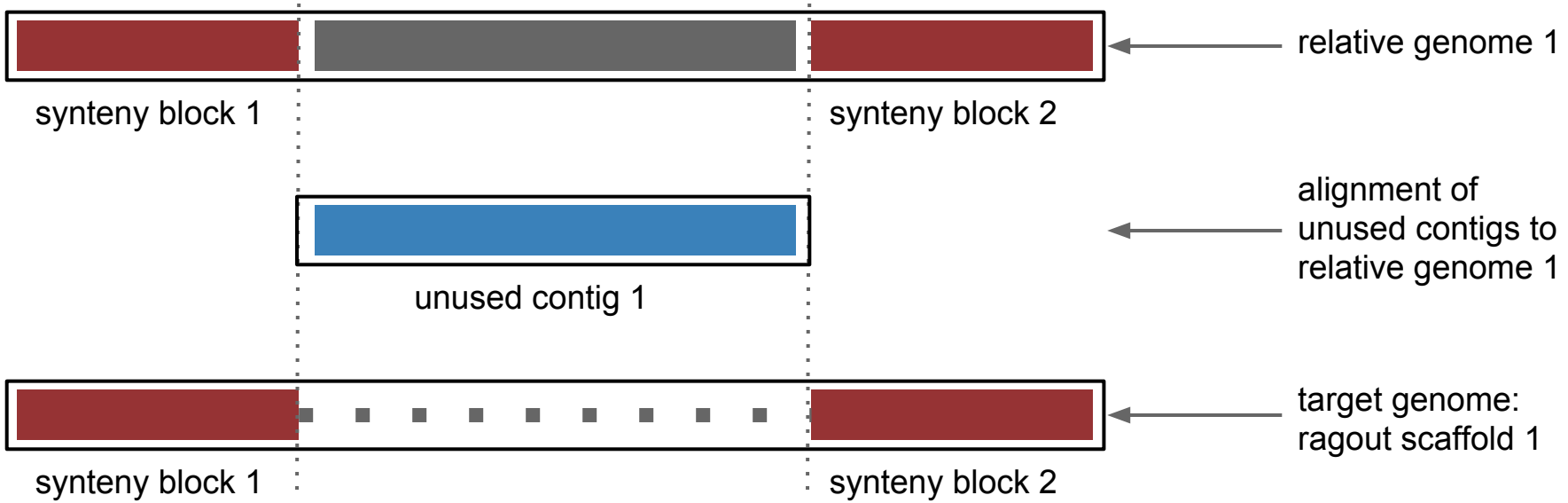
Plan

- Ragout overview
  - Datasets
- Assembly improvements
  - Quality
    - overlap graph
    - paired-end reads
  - **Coverage**

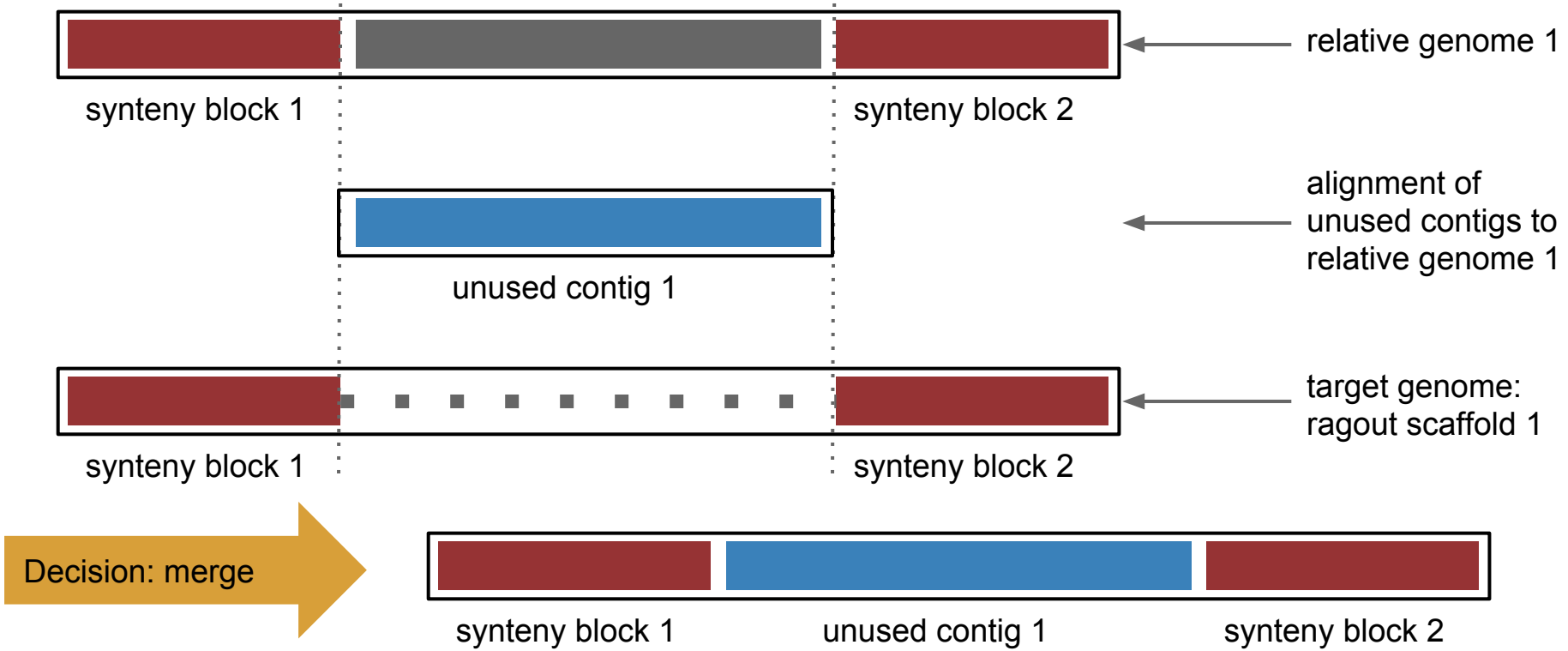
Improve coverage

- align unused contigs to relative genomes
- add consistent onces to scaffolds

Improve coverage: idea



Improve coverage: idea



Unused contigs: tool overview

- Input:
  - reference genomes
  - scaffolds
  - unused contigs
- Output:
  - improved scaffolds

Currently takes Ragout workdir and reference genomes as input.

Pipeline

- build synteny blocks for scaffolds and references (Sibelia)
- align unused contigs to all references (bwa)
- analyze alignment
- repeat for other block size

Adding contigs

- for every alignment of every unused contig save left and right blocks with distances and orientation
- for every consequent synteny block pair in scaffolds check if such pair was seen in alignment
- if so, add contig with the same distances

Adding contigs: reality

Testing on bacterial datasets: nothing can be added

Current ideas, in progress

- use not only left and right synteny blocks, but save a couple of them with different priority
- include information about contig’s length with every synteny block

Unused contigs: final aim

Stand alone tool, which helps to improve coverage for reference-assisted assembly