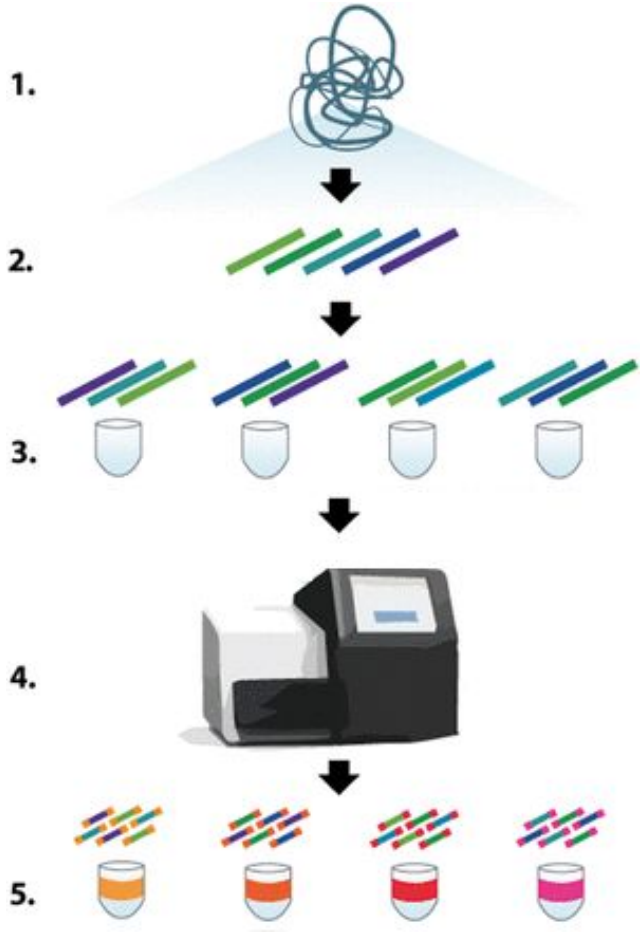
Algorithms for read cloud assembly

Ivan Tolstoganov

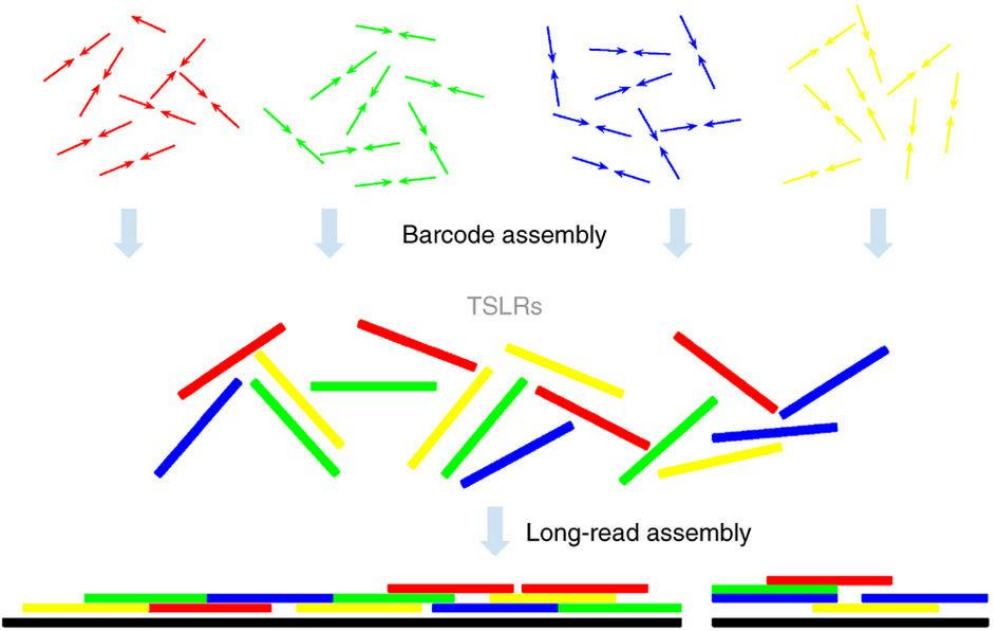
Scientific advisor:
Anton Bankevich

SLRs & Read clouds



1. DNA is sheared into long fragments
2. Fragments are diluted and placed into multiple containers
3. Fragments are amplified, cut into short fragments and barcoded
4. The barcoded fragments are pooled together and sequenced
5. Reads can be demultiplexed into their original compartment via the barcodes in order to form read clouds or SLRs

Synthetic long read assembly

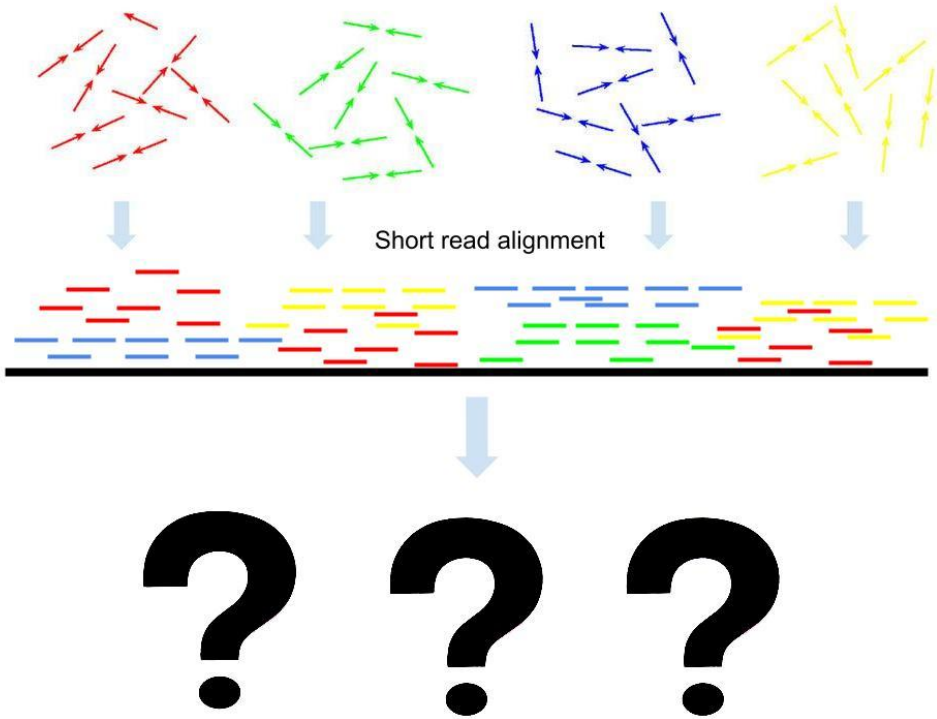


- Each container may be assembled separately to obtain multiple kilobase-long sequences in each well
- The target genome is assembled using the long fragments

Technologies:

- Illumina Tru-Seq SLRs
- Fosmid pooling
- Long fragment reads

Read clouds

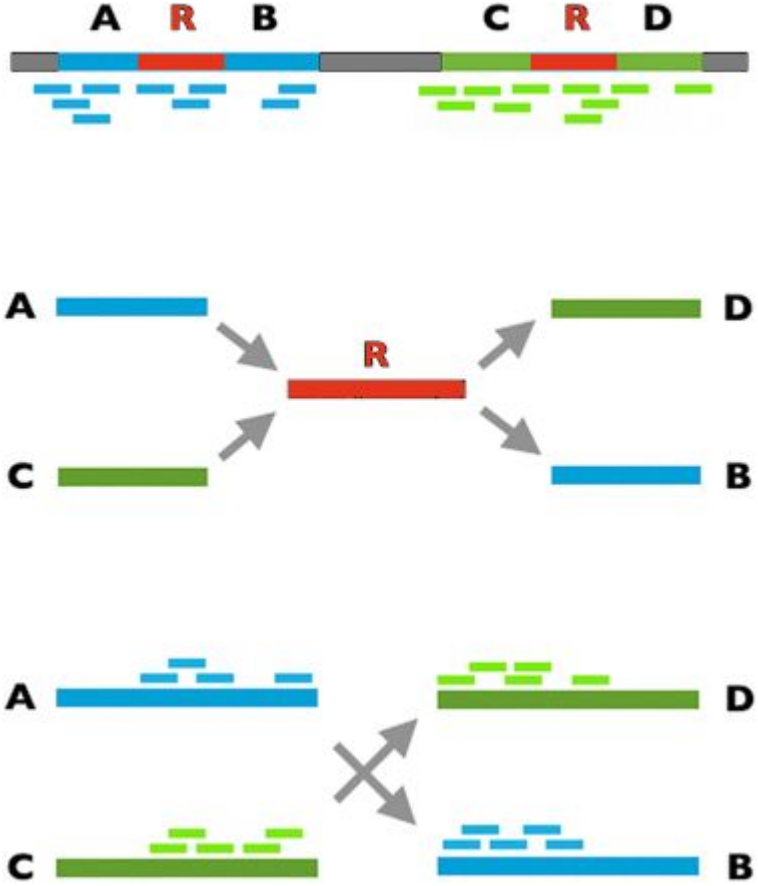


- We may skip subassembly step and obtain clusters of short reads that originate from long fragments
- Such clusters may be referred as read clouds

Technologies:

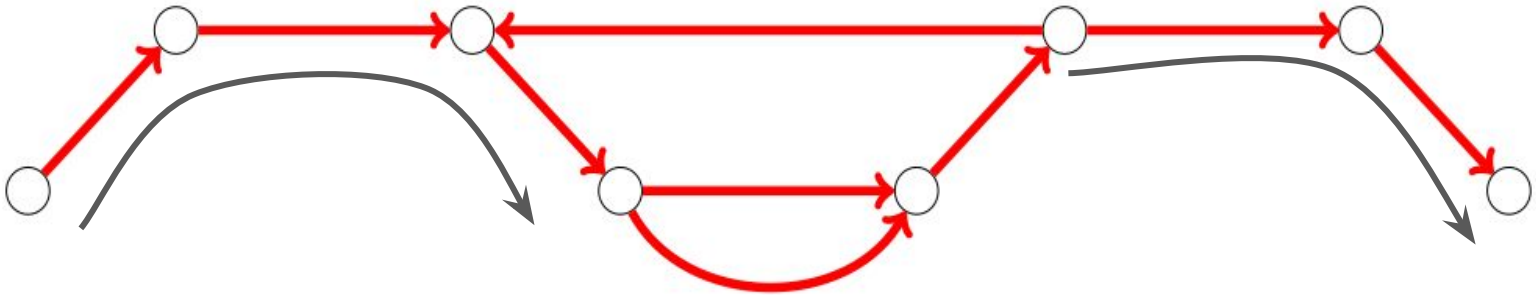
- 10x GemCode
- Contiguity preserving transposase sequencing (CPT-seq)

Repeat resolution using read clouds



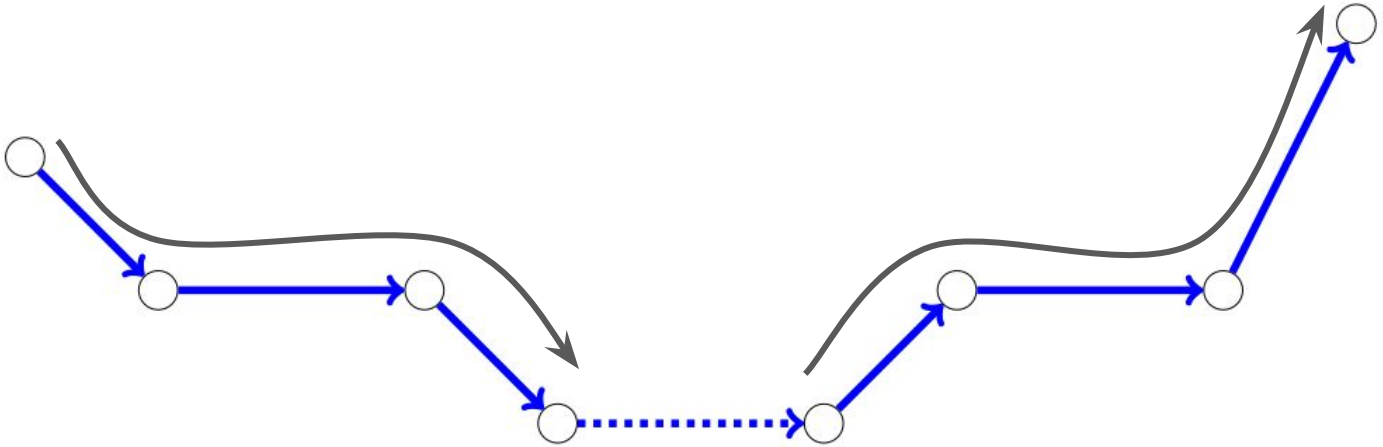
- With short reads, assembly is ambiguous
- Two colored read clusters map, respectively, to ARB and CRD, which may be used to correctly resolve the repeat structure

Read clouds vs SLRs



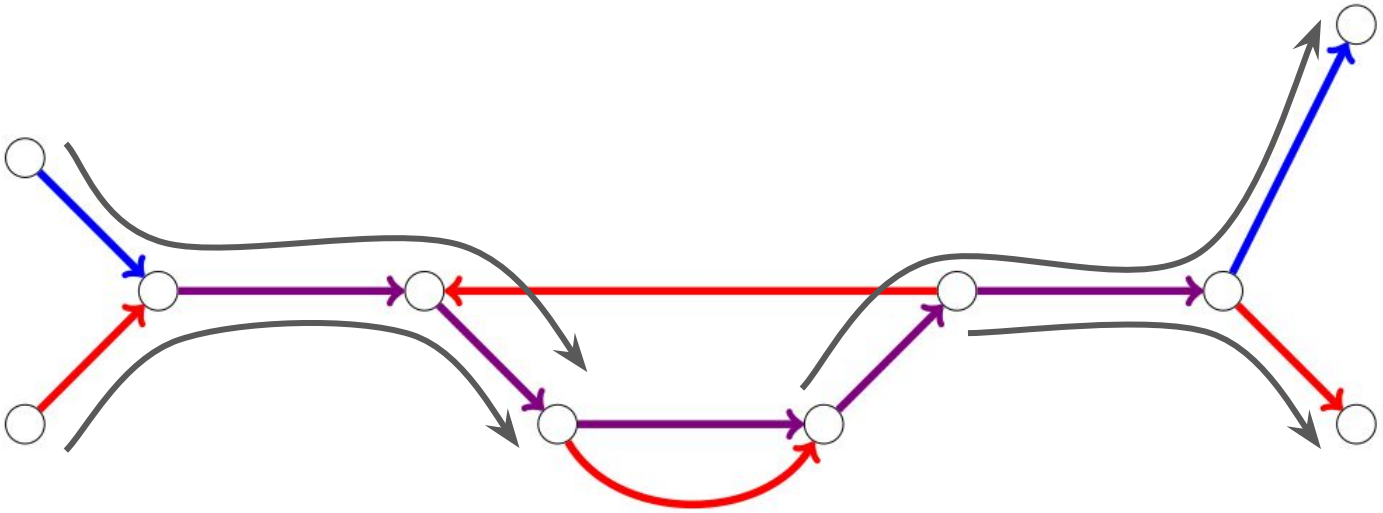
- If long fragment has complex structure, we may not be able to assemble it
- That will result in gaps between same-colored long reads

Read clouds vs SLRs



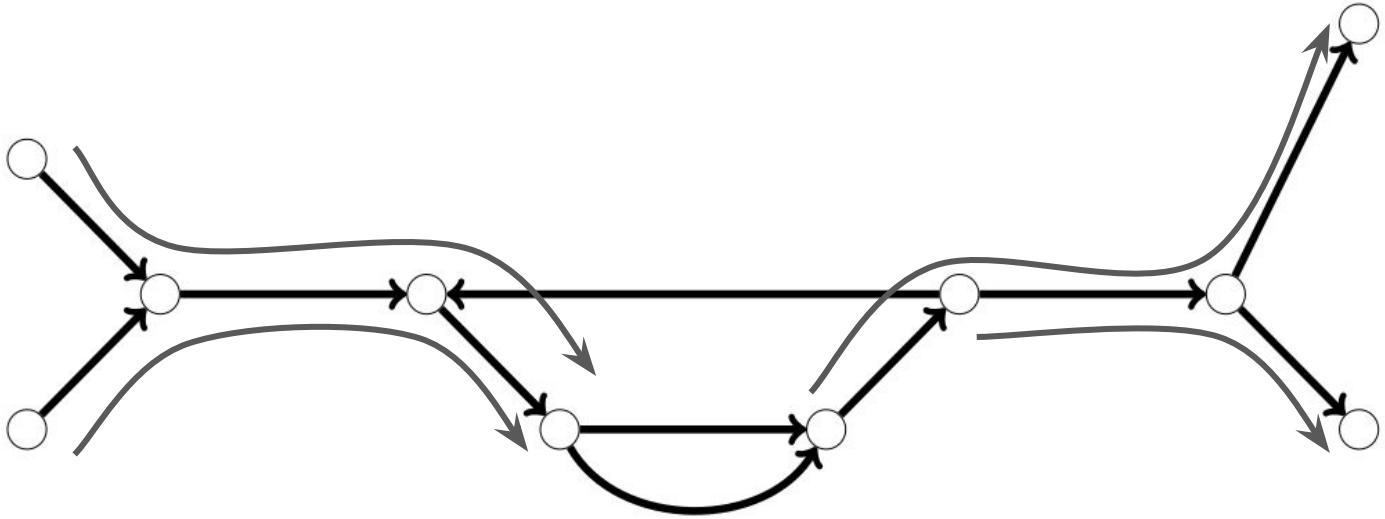
- Same situation occurs with coverage breaks

Read clouds vs SLRs



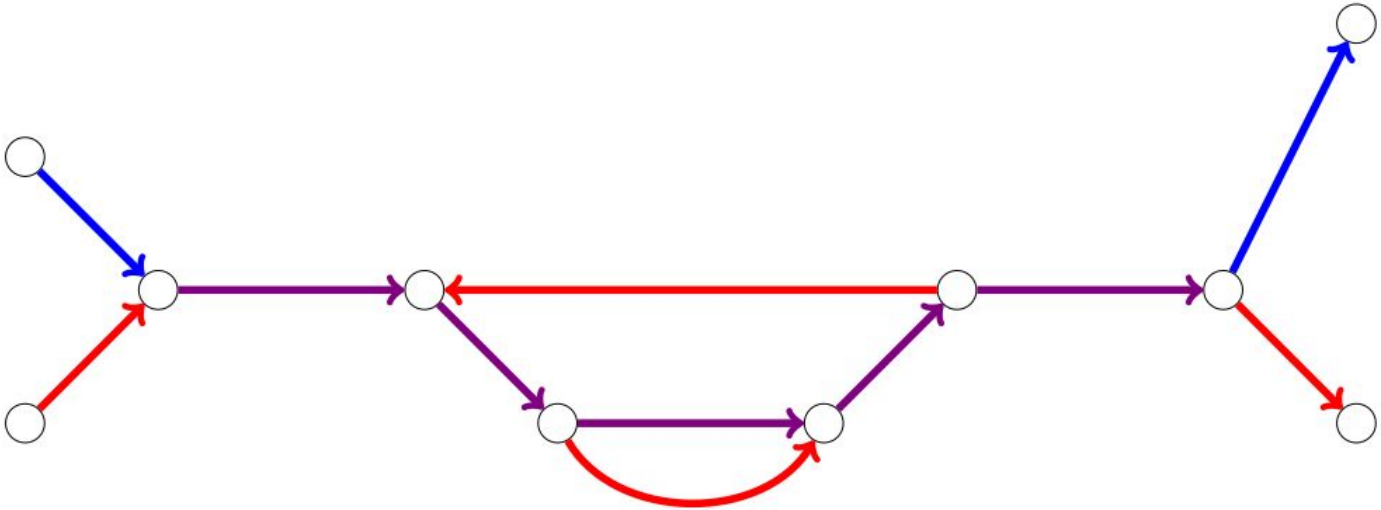
- Consider repeat which consists of two original fragments

Read clouds vs SLRs



- In situations like this repeat resolution using SLRs is problematic

Read clouds vs SLRs



- We can resolve these repeats using barcode information

Read cloud scaffolders

Input:

- Preassembled contigs from a shotgun assembler
- Alignments between the contigs and two sets of reads: paired-end shotgun sequences and read clouds

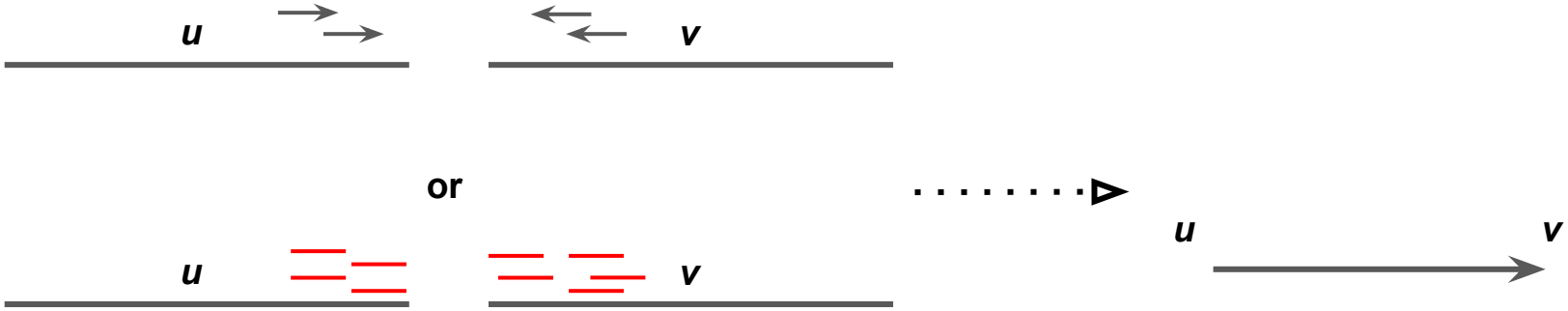
Output:

- Orderings of the input contigs

Tools:

- Architect
- Fragscaff

Architect: graph construction

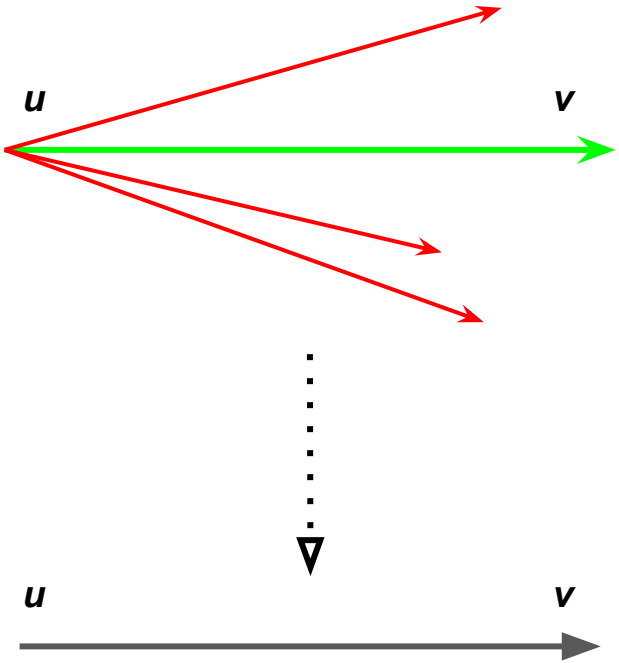


Scaffold graph construction based on

a. number of paired-end links between the scaffolds
b. fraction of shared colors

Kuleshov V, Snyder MP, Batzoglou S. Genome assembly from synthetic long read clouds. *Bioinformatics*. 2016;32(12):i216-i224.

Architect: spurious edges pruning



Edges that have stronger support from given information than all alternatives are identified

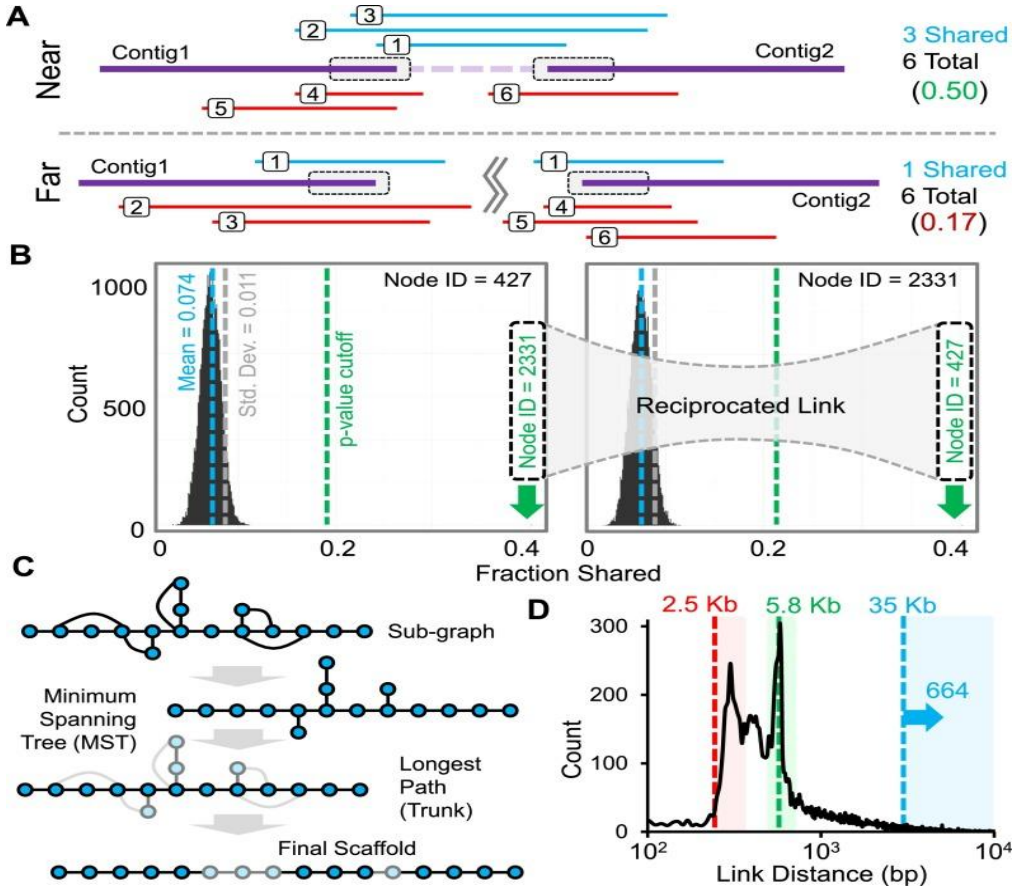
Alternatives are pruned away

There are 3 types of information:

- Paired end
- Read cloud
- Joint paired end and read cloud

Finally, scaffolds are merged if they are connected and there are no alternative connections

Fragscaff



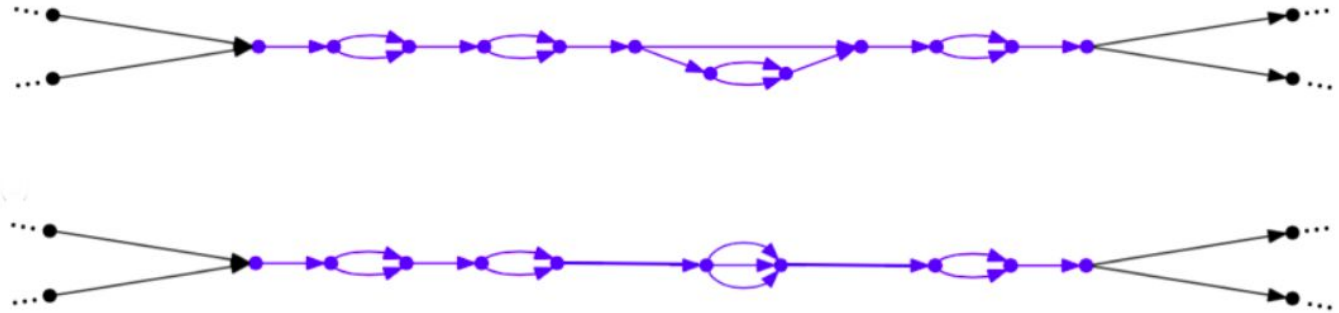
- No paired-end support
- Graph construction is based on the distribution of shared barcodes for each node
- Maximum-weight minimum spanning tree is obtained from each connected component

Adey A, Kitzman JO, Burton JN, et al. In vitro, long-range sequence information for de novo genome assembly via transposase contiguity.

Supernova

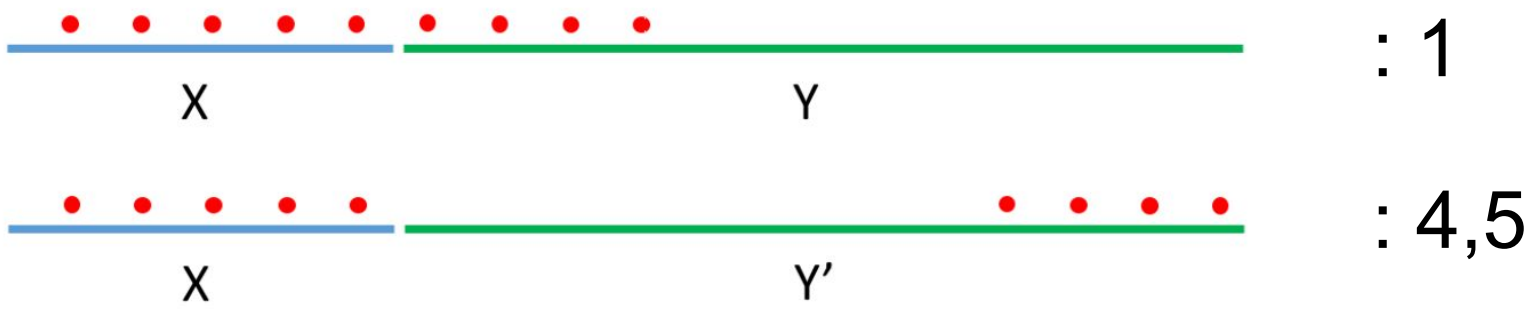
Full scale assembler aimed at 10X Genomics data

- Barcoded reads are used to construct De Bruijn graph
- The graph is decomposed into DAGs with source and sink (“lines”)
- Lines are merged using pair info and barcodes



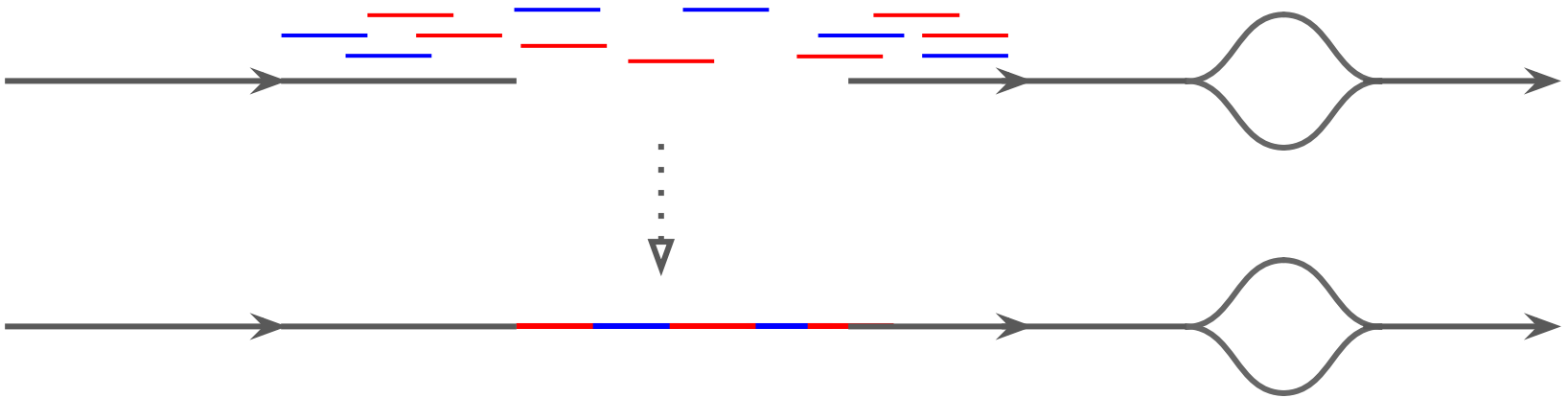
Supernova: ordering

- Set of lines with similar barcodes is collected for every line
- Every possible ordering of this set is scored
- Penalty is counted for every barcode based on distance between read placements for that barcode
- Ordering is treated as a winner if it’s penalty is at least a fixed amount less than that for any competitor



Supernova: scaffolding & phasing

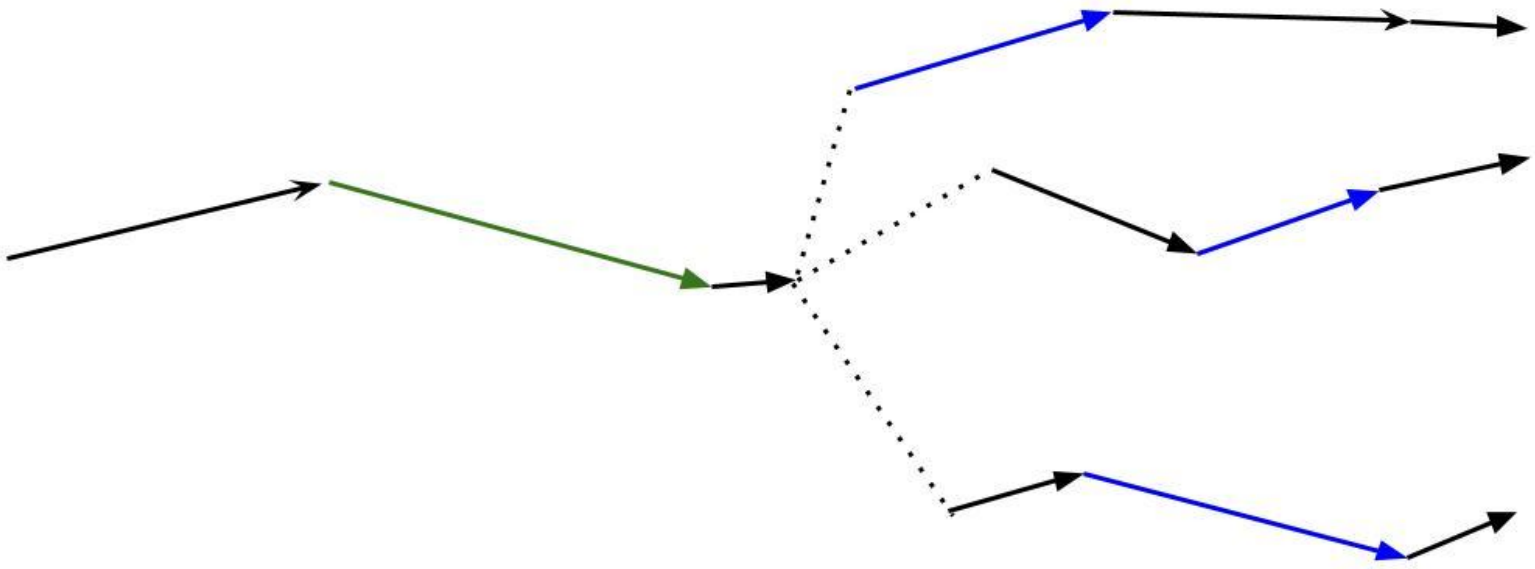
- Short gaps between lines can be removed with pair info
- Long gaps can be replaced by subassembly of reads extracted from shared barcodes at the ends of the gap
- Bubbles with two possible paths are phased by barcode voting



Path extend resolution strategy

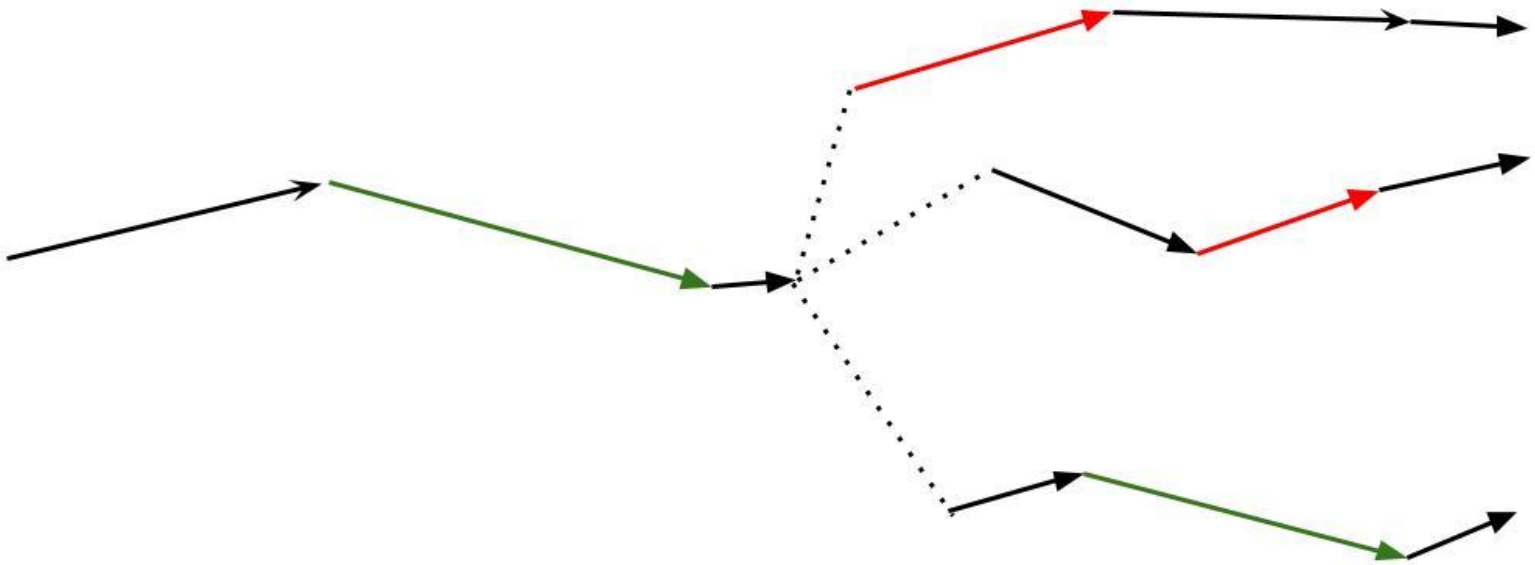
- We take path decomposition of the assembly graph as an input
- Decomposition was obtained from repeat resolver based on paired-end reads
- We intend to join these paths using barcode sets on their edges

Path extend resolution strategy



- First, we should find long edge at the end of the path
- Then we mark long edges that are close to the end of the path as candidates

Path extend resolution strategy



- Provided there is only one edge with sufficient score, we select it as the next edge in our path
- Resulting gaps may be closed via paired end information

Thank you!