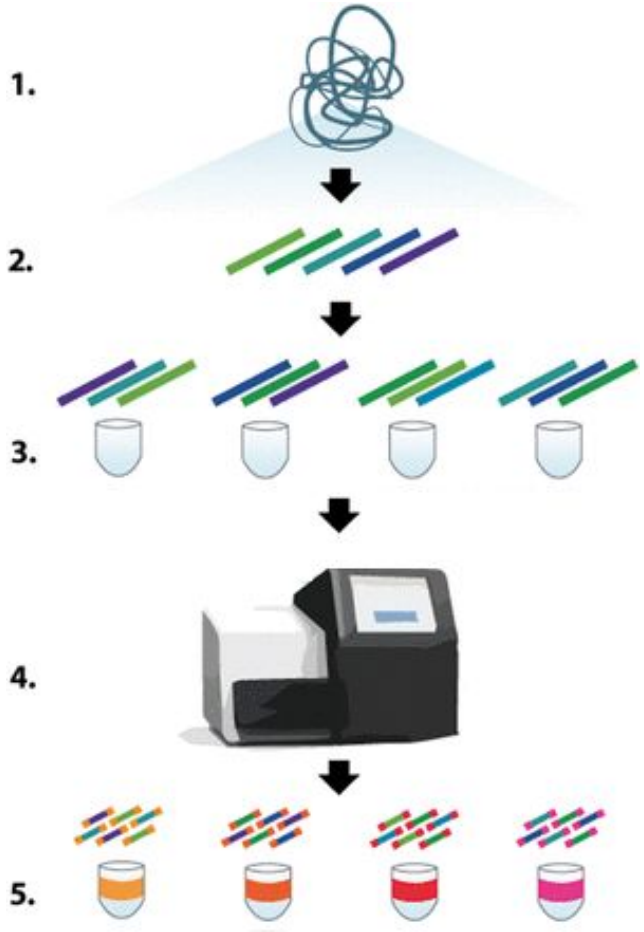
Algorithms for TSLR assembly

Иван Толстоганов

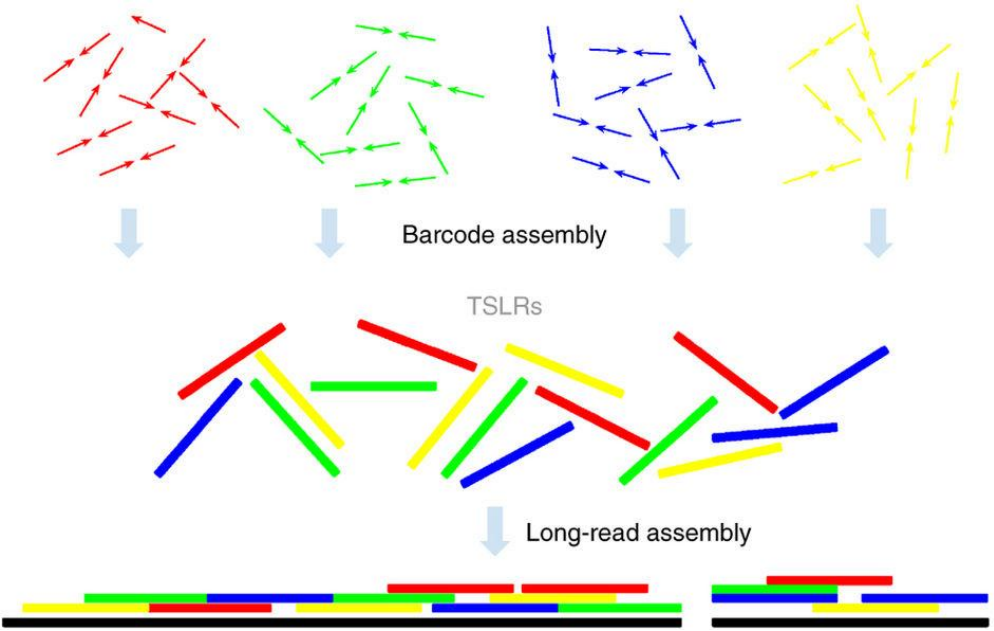
Руководитель: Антон Банкевич

SLRs & Read clouds



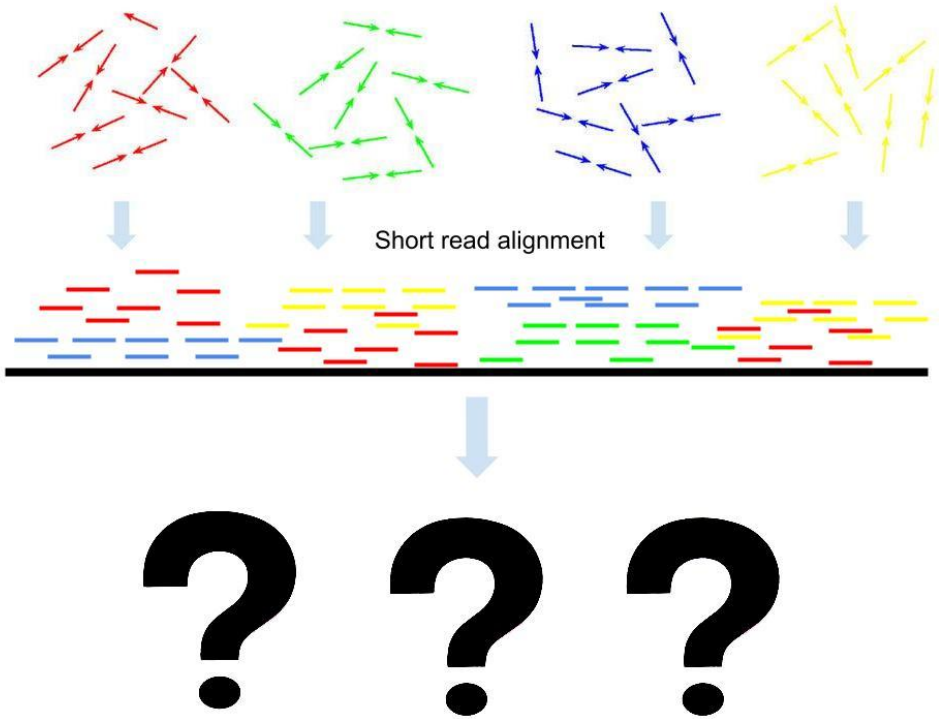
1. DNA is sheared into kilobase-long fragments
2. Fragments are diluted and placed into multiple containers
3. Fragments are amplified, sheared into short pieces and barcoded
4. The barcoded pieces are pooled together and sequenced
5. Resulting reads can be demultiplexed into their original compartment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

Long read assembly



- Each container may be assembled separately with a short-read assembler, which produces multiple kilobase-long sequences in each well
- The target genome is assembled using the long fragments

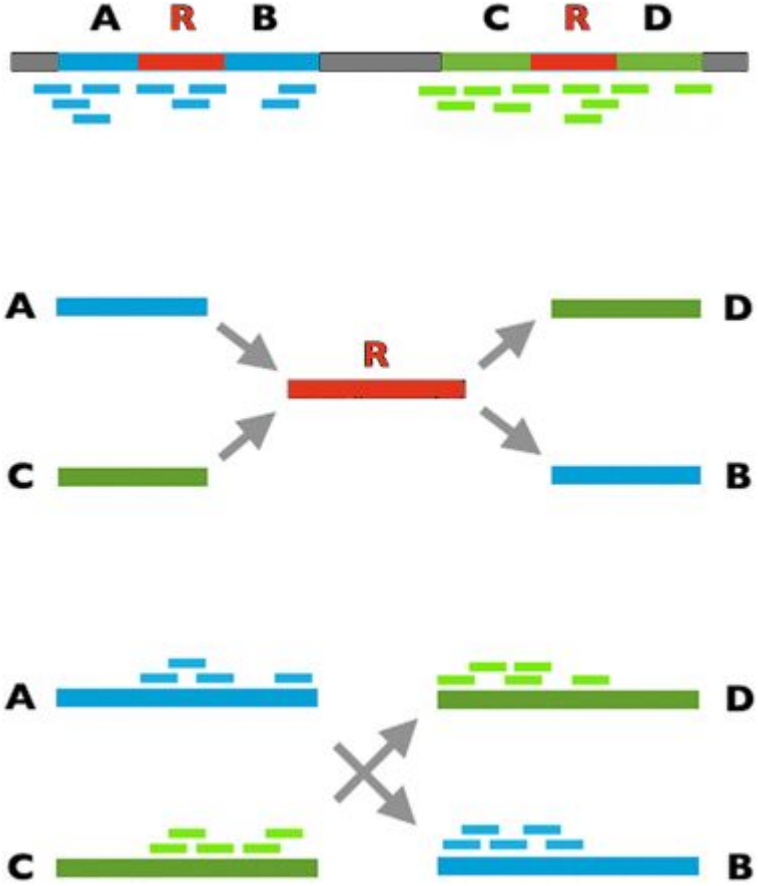
Read clouds



- We may skip subassembly step and obtain clusters of short reads that originate from long fragments
- Such clusters may be referred as read clouds
- Read cloud-based Architect[1] scaffolder has recently been introduced

1) Kuleshov,V. (2016) Genome assembly from synthetic long read clouds. Bioinformatics, 32(12):i216-i224.

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

Project goals

- Analyze existing strategies for TSLR assembly
- Show that strategy based on graph coloring might be more efficient in certain cases
- Implement this strategy and analyze the results

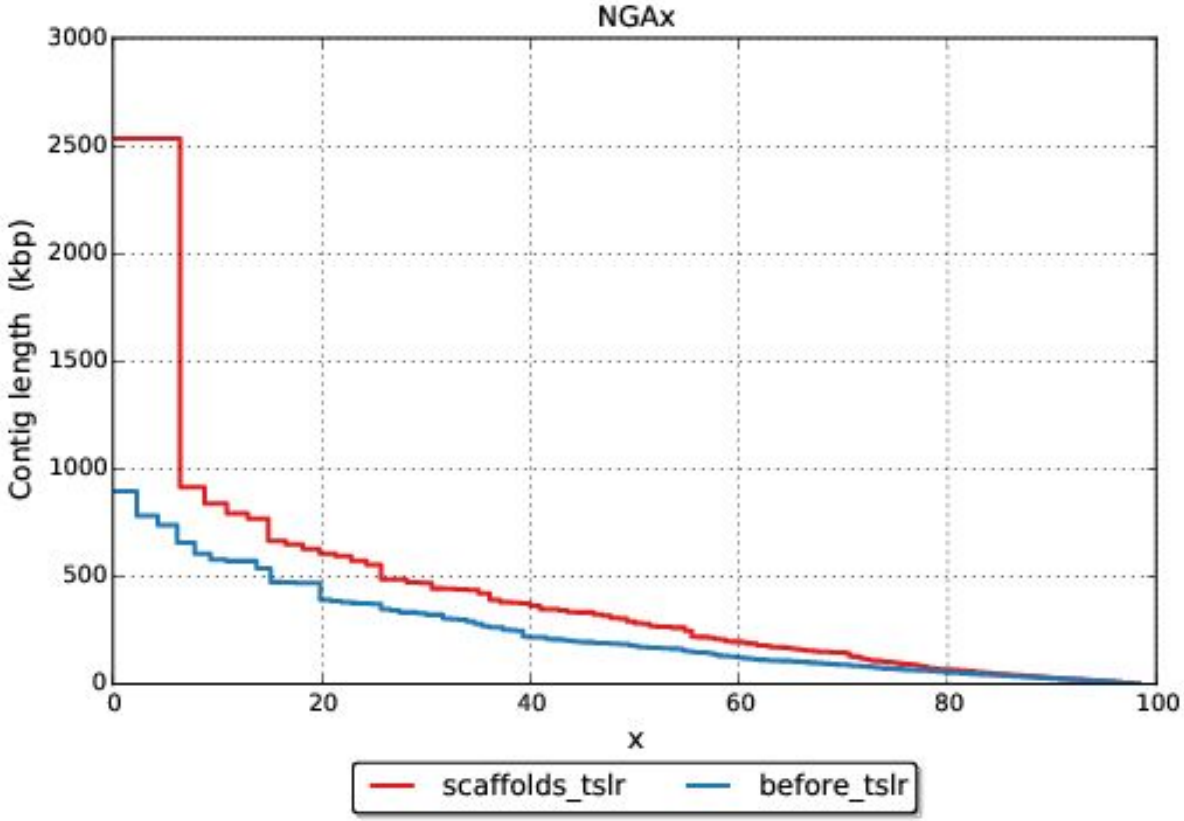
Current repeat resolution strategy

- 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

Results

- Method was verified on the human microbiome synthetic community
- Assembly graph was constructed from ideal paired-end library
- N50 increased from 185 kbp to 310 kbp compared to contigs from paired-end resolver
- Very few additional misassemblies were made

Results


NGAx
Contig length (kbp)
3000
2500
2000
1500
1000
500
0
0
20
40
60
80
100
x
scaffolds_tslr
before_tslr

Main challenges

- High coverage appeared to be the most notable problem
  - Overall number of barcodes is limited to 384
  - Therefore, edges with large number of barcodes can’t provide reliable information
  - We can’t find a difference between random and real barcode matches
- Current resolver is very dependent on the assumption that certain edges are unique
- Works very slow on well-connected assembly graphs, such as those that are built from mammalian datasets

Future plans

- Devise strategies to deal with high-covered edges
- Make a faster version of resolver to work with human data
- Combine with repeat resolution for subassembled TSLRs