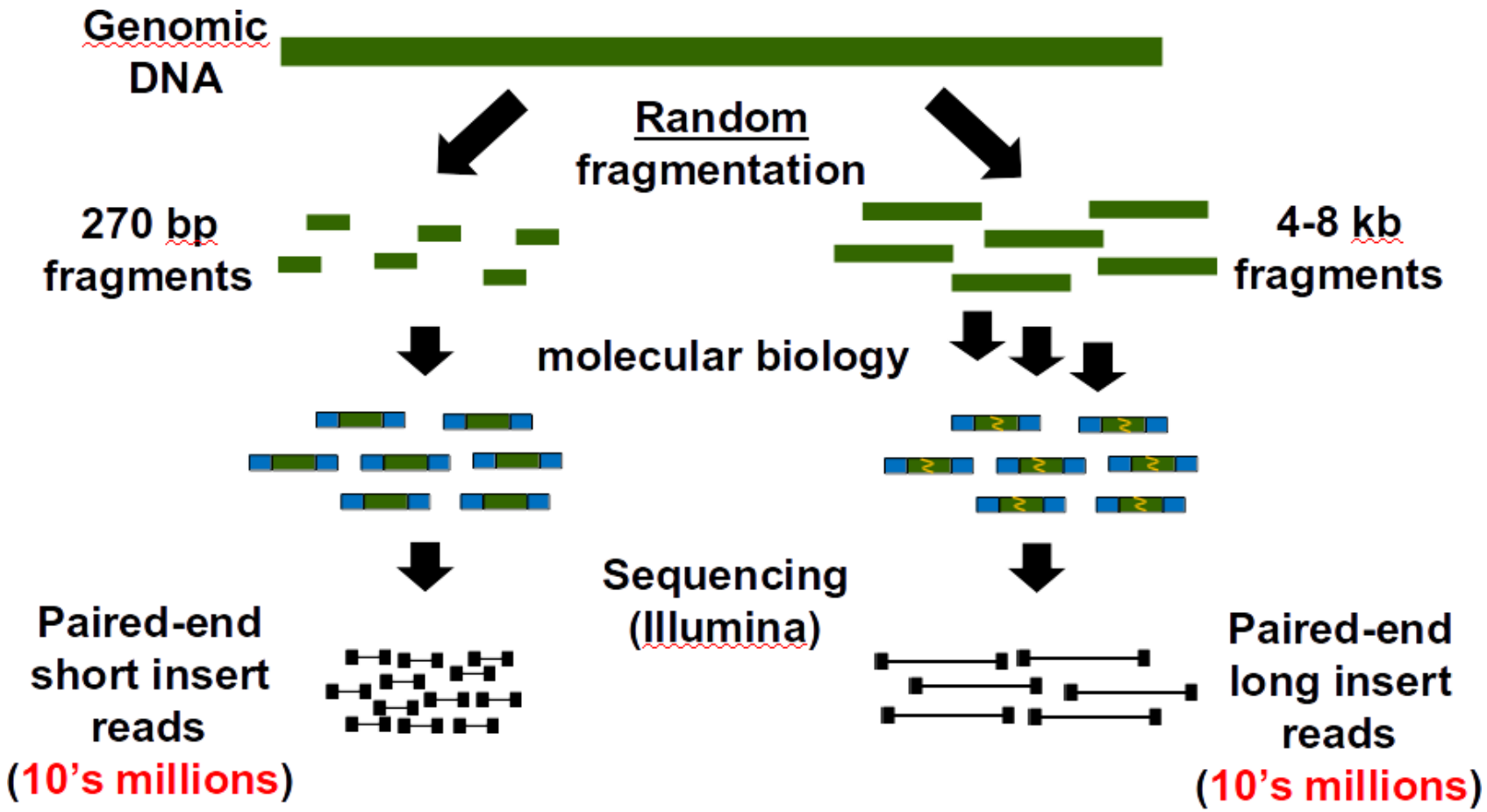
Lab 3: Short-read assembly: An overview

Maoying, Wu

What is assembly?



Terminology and definitions

Fragment library a shrot insert (270bp) library with overlapping ends, a.k.a standard library

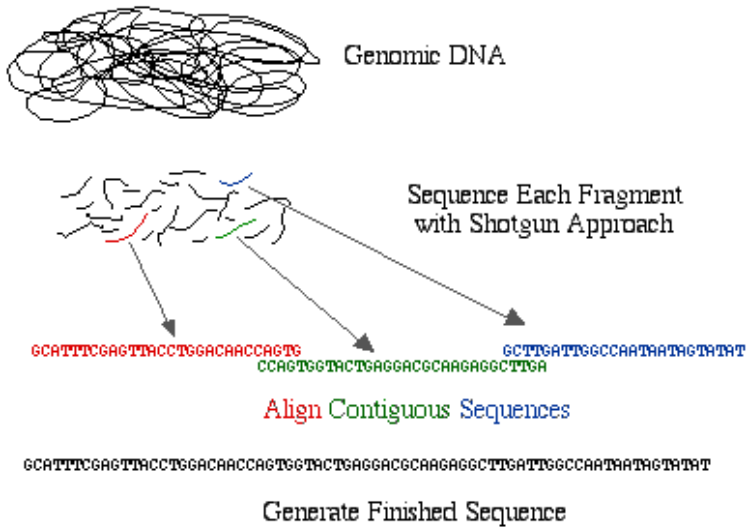
Long insert library A 4-8kb librbary where only 100 bps on each end are sequenced. a.k.a CLIP, mate pair library

Contig A contiguous sequence of DNA

Scaffold One or more contigs linked together by unknown sequence

Captured gap A gap within a scaffold. The order and orientation of the contigs spanning the gap is known.

Traditional approach

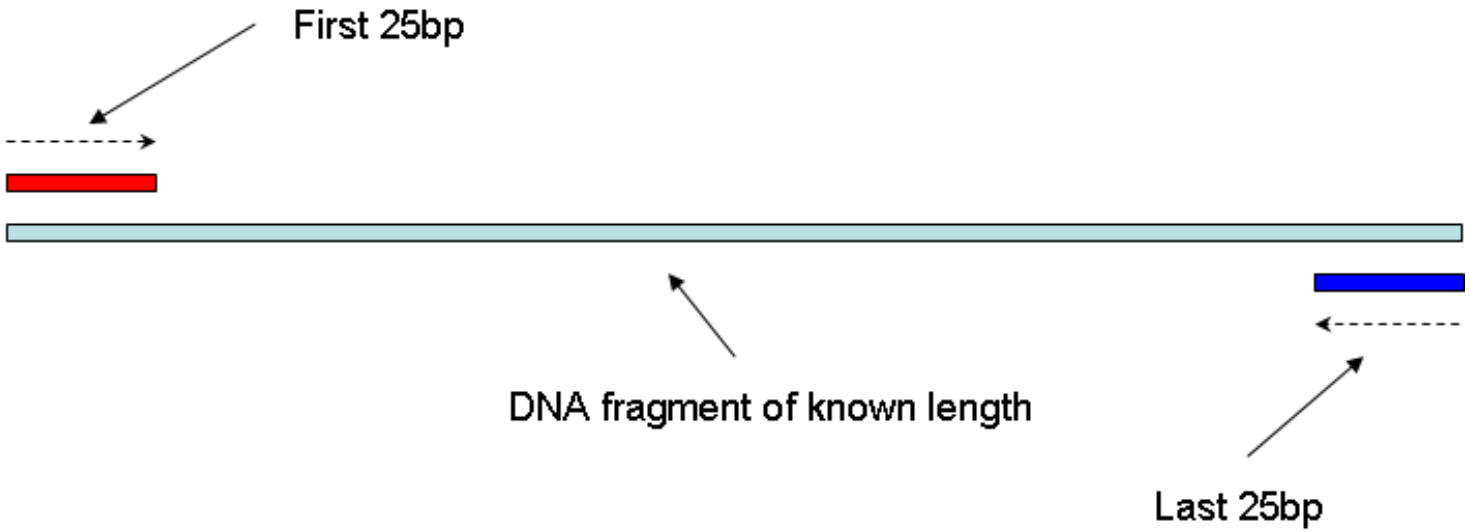


- Sanger capillary sequencing often produce fragment of length 600 bp.
- $\sim 10x$ coverage / sequencing depth
- Assebled using **overlap-layout-consensus** approach.
  - Build an overlap graph where each node represents a read. An edge exists between two reads iff they overlap.
  - Traverse the graph to find unambiguous paths which form contigs.

Next generation sequencing (NGS)

- Roche/454, Illumina Solexa, ABI SoLID
- Much higher throughput
- Lower cost
- Very short fragment lengths (25-300bp)
- Higher error rate
- Single-end or paired-end (mate-pair) sequencing

Paired-end sequencing



- Sequencing two ends of a fragment of known size
- Currently fragment length (insert size) can range from 200 - 10,000 bps

Assembly for NGS: Challenges

Challenges: *impractical overlap-layout-consensus*

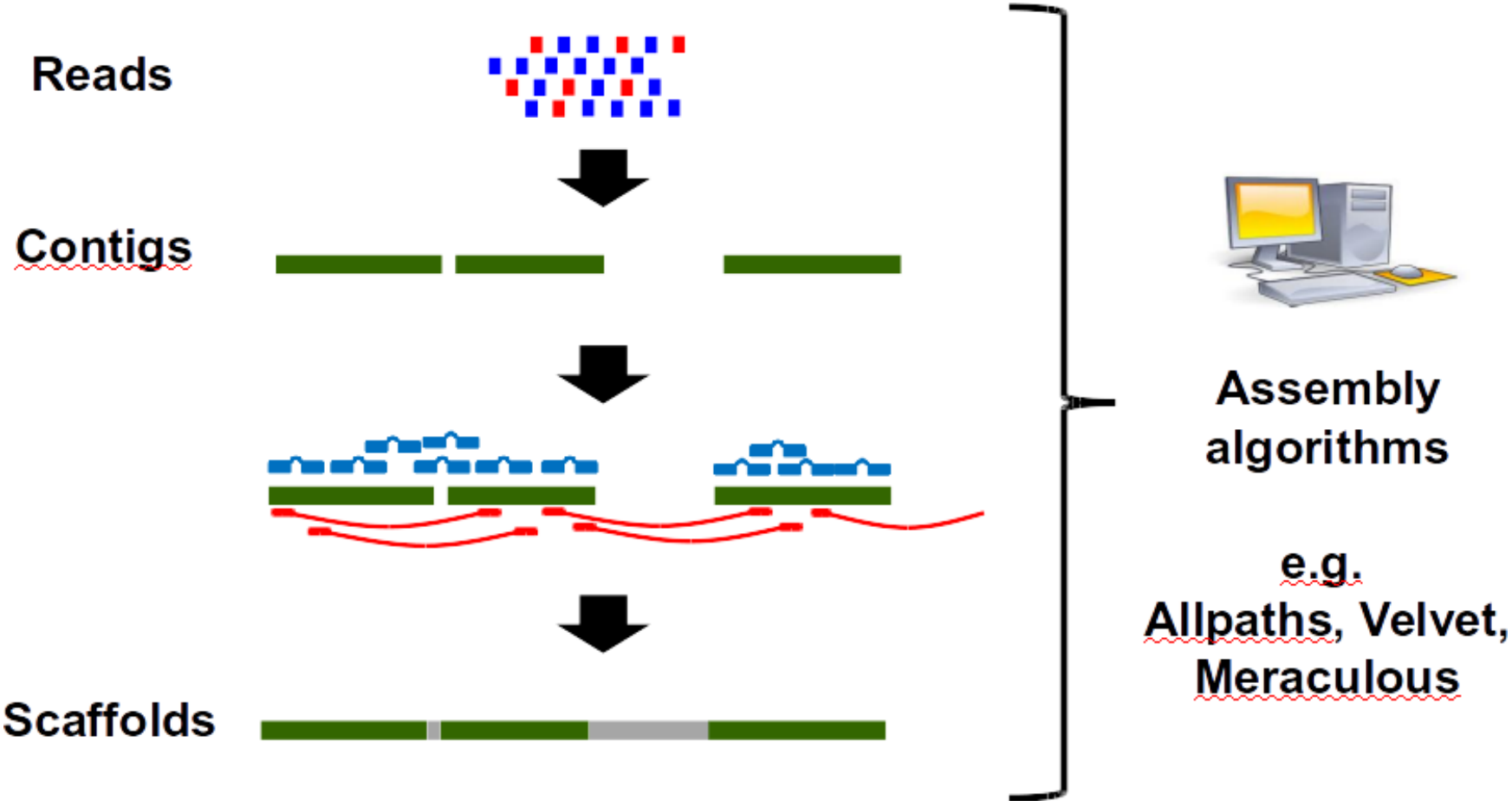
- $> 100x$ coverage, result in *a large number of reads*
- Short fragment $\rightarrow$ *short overlaps* will generate large fraction of false overlaps.
- *High sequencing error rate*

Solutions:

- de Bruijin graph assembly
- Seed and extend assembly

Velvet: the most popular short read assembly method.

Current approach



Genome assembly programs

Name	Algorithm	Data
Abyss	De Bruijn	Illumina
Allpaths-LG	De Bruijn	Illumina/PacBio
CABOG (Celera)	OLC	All
HGAP	OLC	PacBio
Masurca	De Bruijn/OLC	All
Mira	OLC	All
Newbler	OLC	454/Illumina/Torrent
SGA	String	Illumina
SoapDeNovo	De Bruijn	Illumina
Spades	De Bruijn	Illumina
Velvet	De Bruijn	Illumina

De Bruijn graph method

Pros

- high efficiency

Cons

- Partitioning all reads into k-mers is memory- intensive. For mammalian genomes, 1000+ GB RAM is prohibitively expensive.
- Loss of connectivity information between reads due to the chop of reads into k-mers. The longer the reads, the more information is lost.

Assembly tips

- The structure of the graph is highly dependent on the *k*-mer size used for assembly.
  - Small *k*-mers result in shorter contigs but with lots of connections.
  - Large *k*-mers can result in longer contigs but with fewer connections.
- If your graph consists of many separate disconnected subgraphs, then your *k*-mer size may be too large.
- If your graph is connected but is very dense and tangled, then your *k*-mer size may be too small.
- When assembling 100 bp reads in Velvet, a *k*-mer of 51 would be a good starting point, and then adjust up or down as needed.
- SPAdes conveniently conducts assembly multiple times using different *k*-mers, so you can look at the FASTA files for each assembly (in folders named like K21, K33, etc.) to find the best graph for viewing in Bandage.

Bandage: A GUI program to view assembly graphs

- *De novo* assembly graphs contain assembled contigs (nodes) but also the connections between those contigs (edges).
- Bandage visualizes assembly graphs, with connections, using graph layout algorithms.
- Users can interact with the graph by moving, labeling and coloring nodes.
- Sequence information can be directly extracted from the graph viewer.

Install Bandage

1. sudo apt-get update
2. sudo apt-get install build-essential git qtbase5-dev libqt5svg5-dev
3. Prepare the OGDF library:
   - Download OGDF code from `http://www.ogdf.net` and unzip
   - Create the Makefile: `./makeMakefile.sh`
   - Compile the library: make
4. Download the Bandage code: `git clone https://github.com/rrwick/Bandage.git`
5. Set the environment variable: `export QT_SELECT=5`
6. Run qmake to generate a Makefile: `qmake`
7. Build the program: `make`
8. You can install both OGDF and Bandage to `/usr/local/bin`
9. Run Bandage

SGA: String graph assembler

- a.k.a. Simpson-Durbin assembler
  (1) computes overlaps between all read pairs
  (2) constructs a string graph based on the overlaps.
  (3) derives the genome assembly from the string graph.
- Pros:
  - considerably lower memory demand
  - handle huge genomes at a significantly lower cost.

  Of the four assembler, SGA used the least memory (4.5 GB vs. 14.1 GB, 23.0 GB and 38.8 GB for ABySS, Velvet and SOAPdenovo, respectively)

SGA: Reference

String Graph concept:

- E. W. Myers. The Fragment Assembly String Graph.