Genome assembly and annotation

Day 3: Genome assembly

Antti Karkman

Department of Microbiology – UH

antti.karkman@helsinki.fi

Slides modified from Igor Pessi, with permission

Aims for this part of MMB-114

Day 1: Basics of UNIX and working with the command line

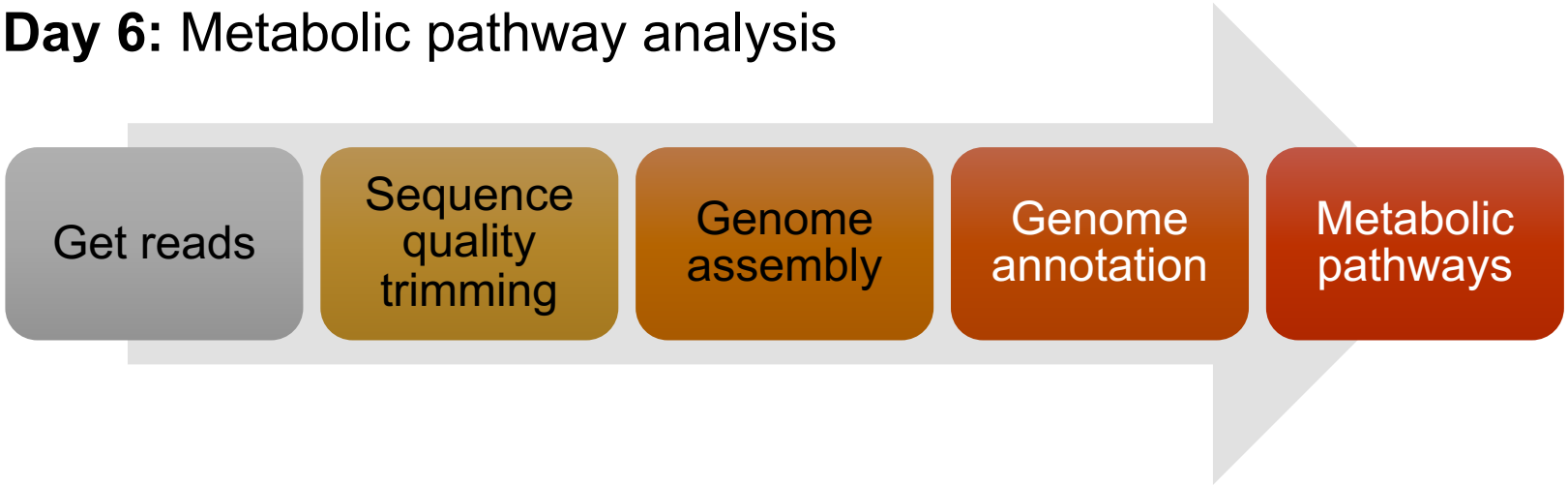
Day 2: Handling of Illumina data

Day 3: Genome assembly

Day 4: Check-up and report

Day 5: Genome annotation

Day 6: Metabolic pathway analysis



Before we start...

How does the raw data look like?

How many sequences we have for each genome? Are these numbers the same for the R1 and R2 files?

Is the length of the reads the same in the R1 and R2 files?

The "Per base sequence quality" module shows a problem. Why?

- Which part of the reads have the best quality? Beginning, middle, end?

What is the mean sequence quality for the majority of the reads?

The "Per base sequence content" module shows a problem. Why?

The "Adapter content" module also shows a problem. Why?

- Are there adapters in our reads?
- In which part of the reads?
- What are adapters and why should we remove them?

Which of the three genomes has the best sequence quality?

How does the data look like now?

CUTADAPT

How many times adapters were trimmed in R1 and R2?

How many reads were removed because they were too short?

How many low-quality bases were trimmed?

FASTQC

Do the "Per base sequence quality" plots look different now? Why?

What is the mean sequence quality for the majority of the reads now?

The "Sequence length distribution" module shows a warning now. Have the read lengths changed? Why?

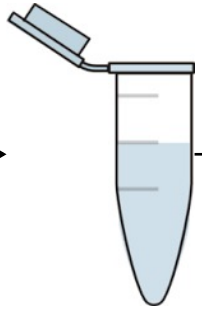
Are there still adapter sequences in our reads?

De novo genome assembly

In an ideal world:

Bacteria

DNA

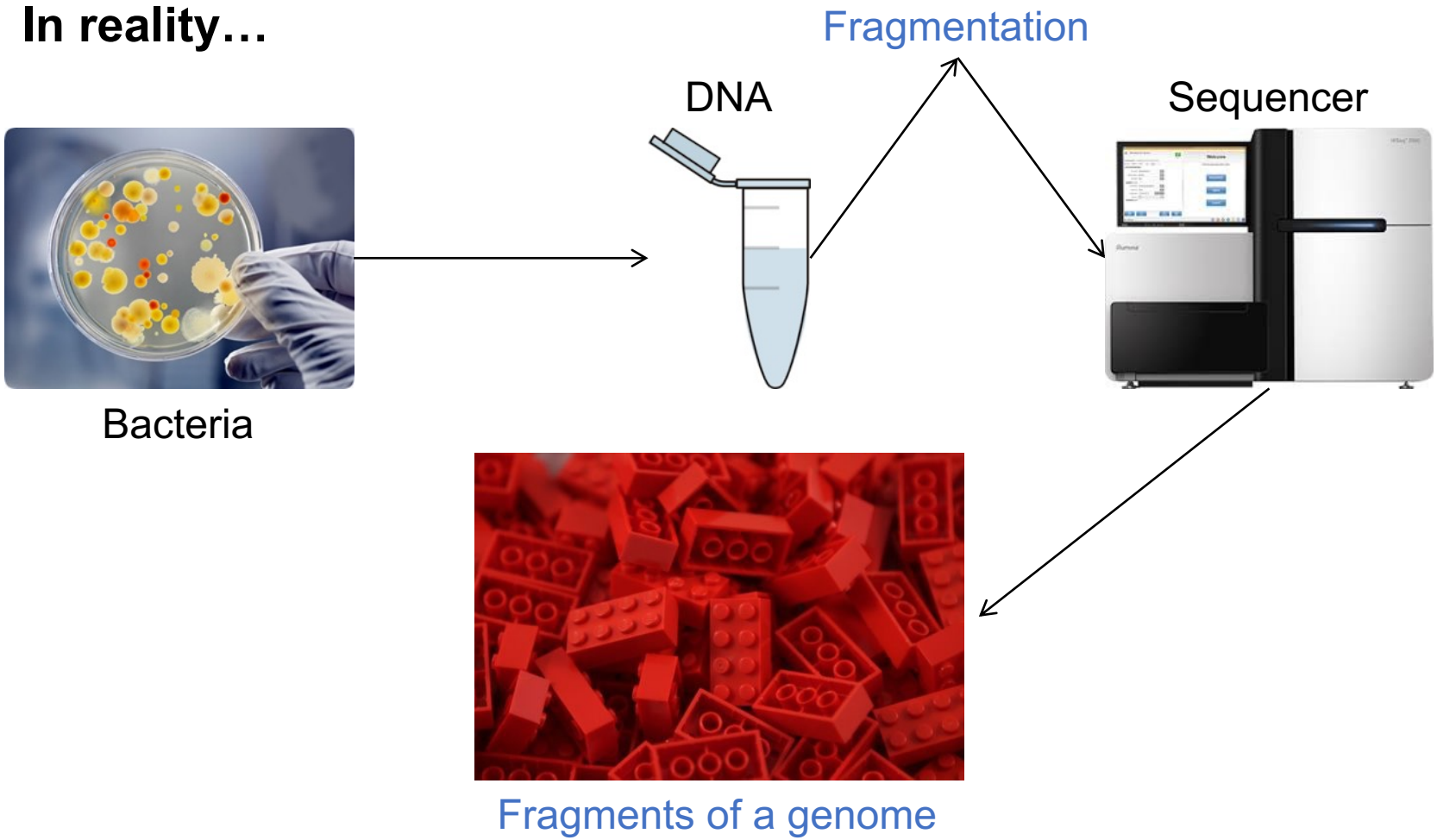


Sequencer

Genome sequence

De novo genome assembly

In reality…



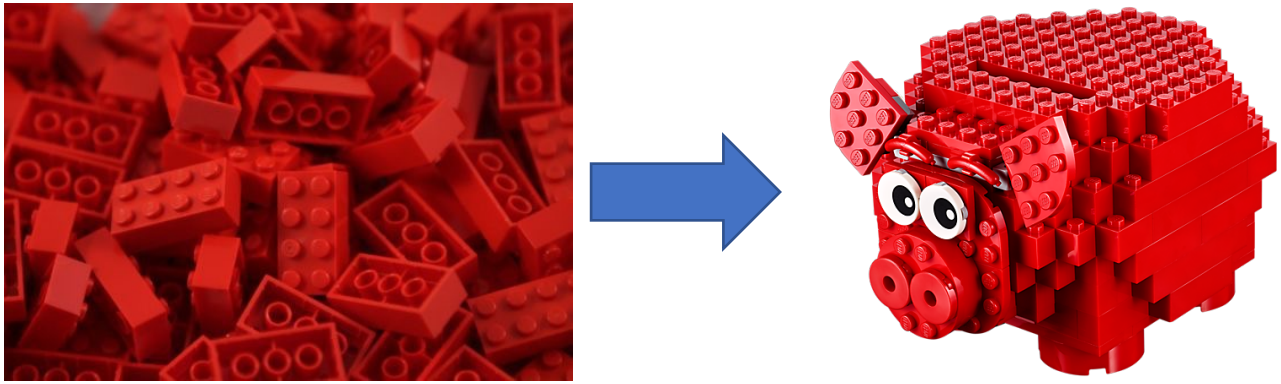
Genome assembly

Reconstruct the original genome from short sequence reads

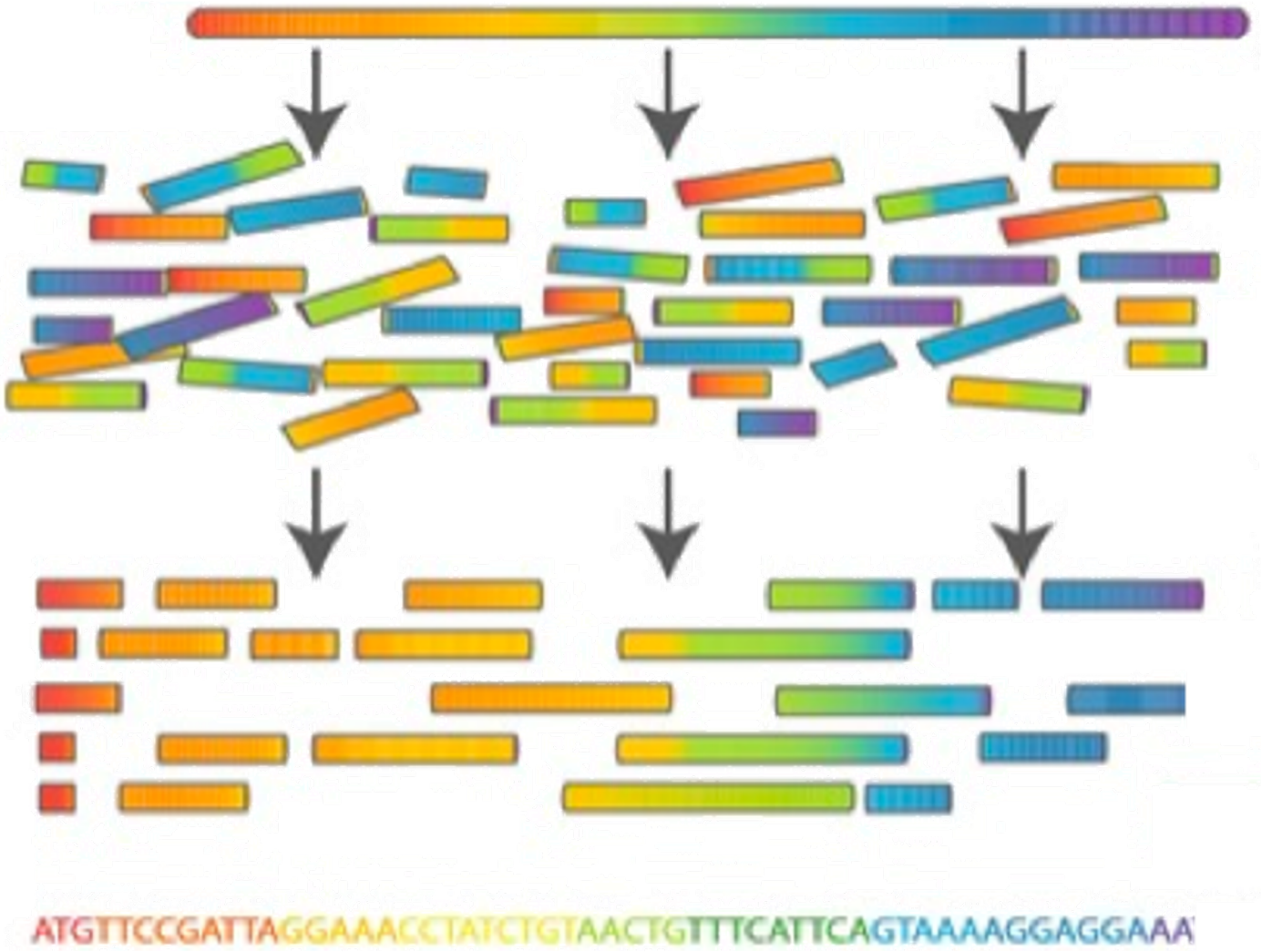
In an ideal world: sequences = one complete genome

In reality: sequences = multiple contigs

- Contiguous, unambiguous stretches of sequences



Genome assembly



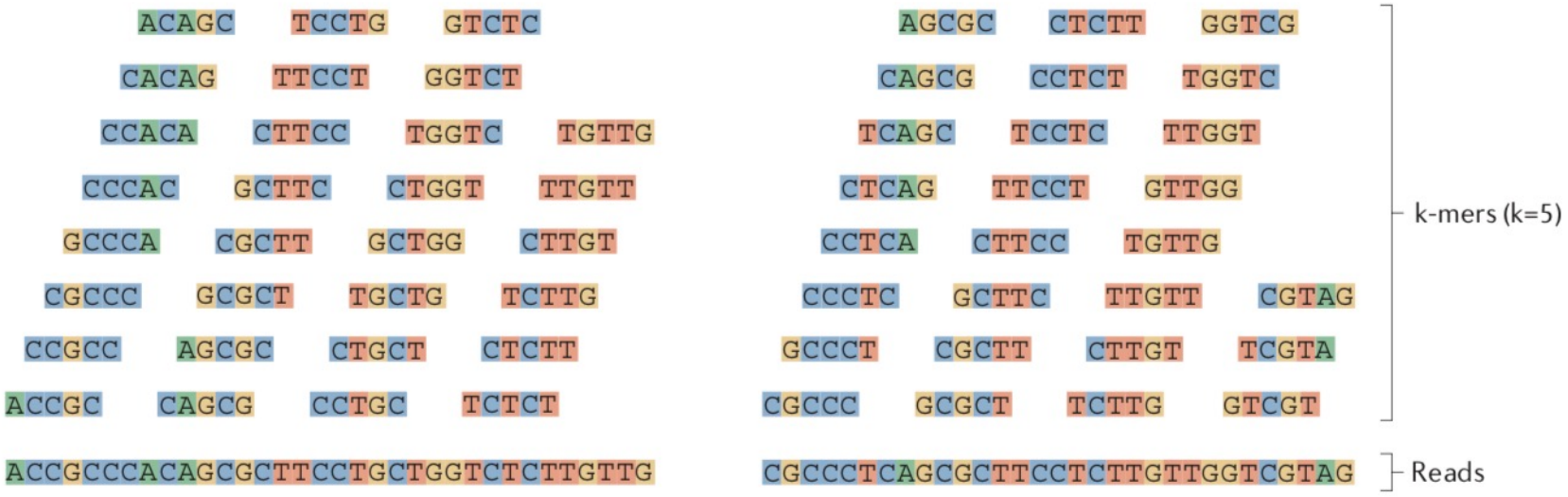
De Bruijn graph

In real life 10^6 sequences cannot be compared with each other

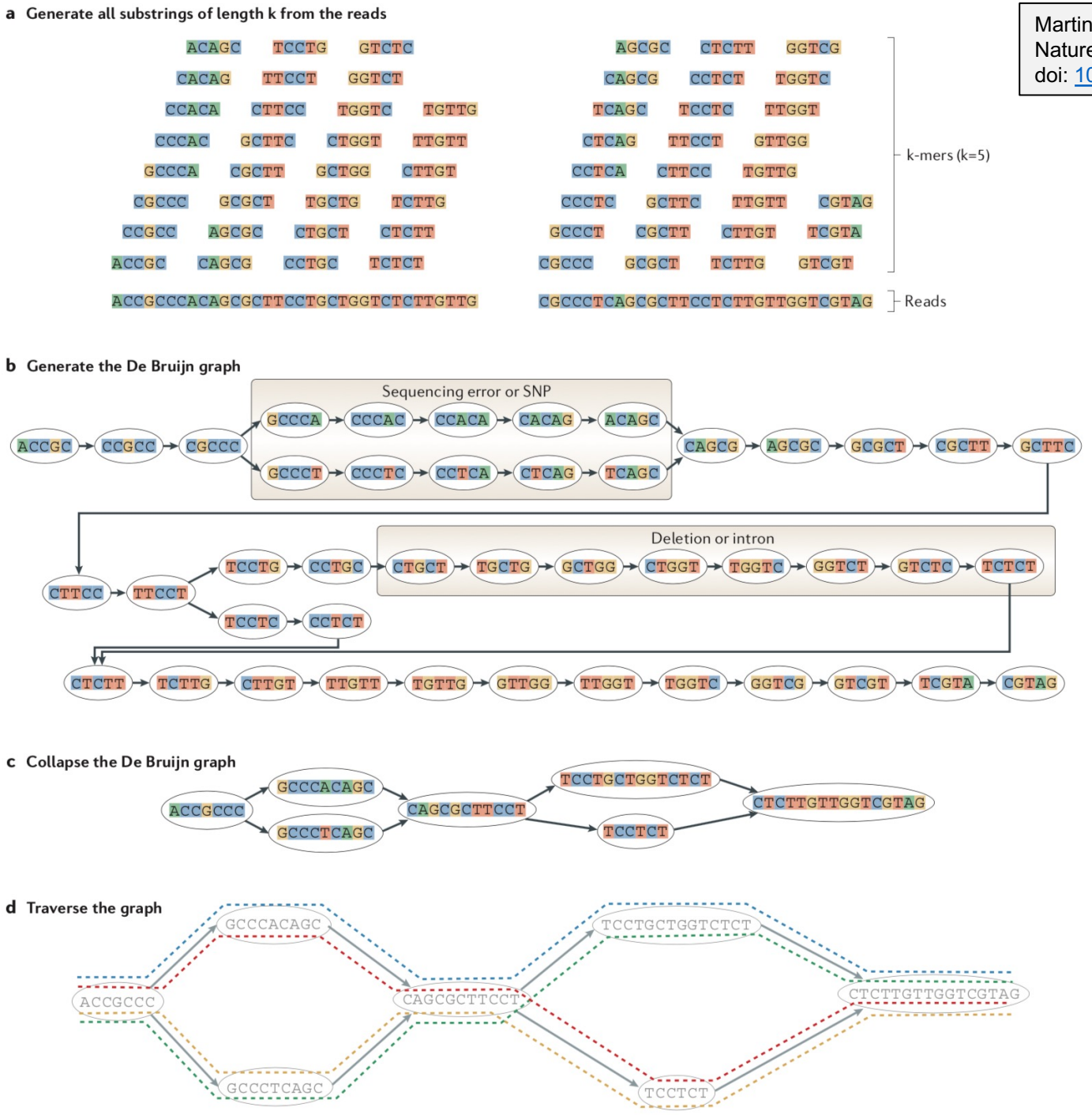
- 10^6 x 10^6 = 10^{12} comparisons

Sequences are reduced to k-mers

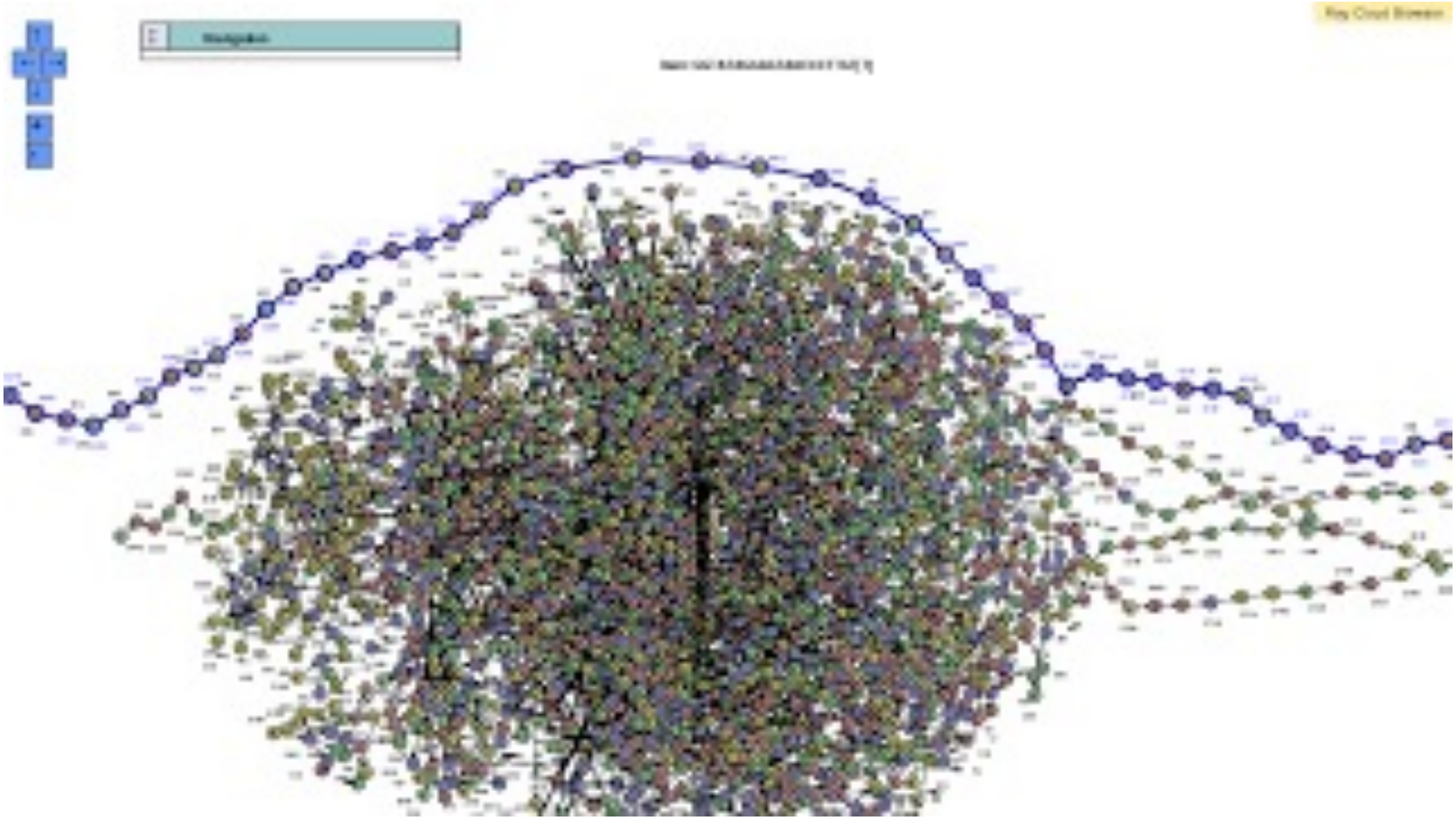
- Smaller subsets of length k



Martin & Wang 2011
Nature Reviews Genetics
doi: 10.1038/nrg3068



In real life with 10^6 sequences



What makes assembly tricky?

Many pieces (computationally-intensive)

Errors in sequence (which one is correct?)

Missing fragments

Repetitive fragments

Multiple copies (rRNA gene as an example)

Circular genome (no starting point)

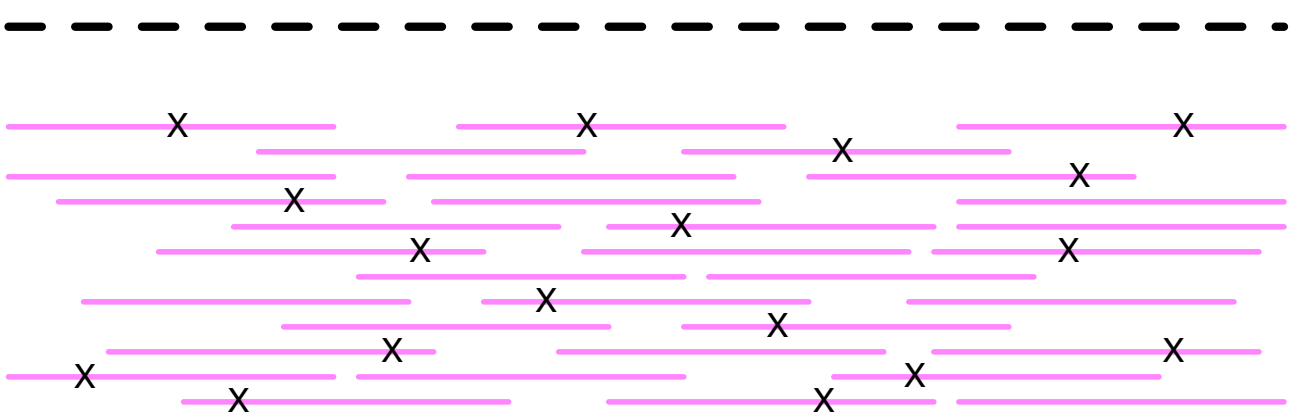
Choice of k-mer

- Too small = misassembly (anything can assemble)
- Too long = no assembly

Sequencing coverage

Coverage describes the number of time that each base of the genome is present in the reads

Assemblers expect equal and high enough coverage (>50x) to work optimally in genome assembly.



Estimated coverage for our *XX* strain

XX genomes: ~2 Mb

Received data

- XX read pairs
- 326 + 278 bp

Coverage

Number of bases sequenced/genome size

Bases = (1,676,086 x 326 bp) + (1,676,086 x 278 bp)

546,404,036 bp + 465,951,908 bp = 1,012,355,944 bp

Coverage = 364,261,528 bp / 2,000,000 bp = **506.2**

Assemblers

Bioinformatic tools that combine short sequencing reads into longer contigs

For our genome, we will use one assembler called SPAdes

- http://bioinf.spbau.ru/spades
- Uses De Bruijn graphs

We will use the batch job system in Puhti

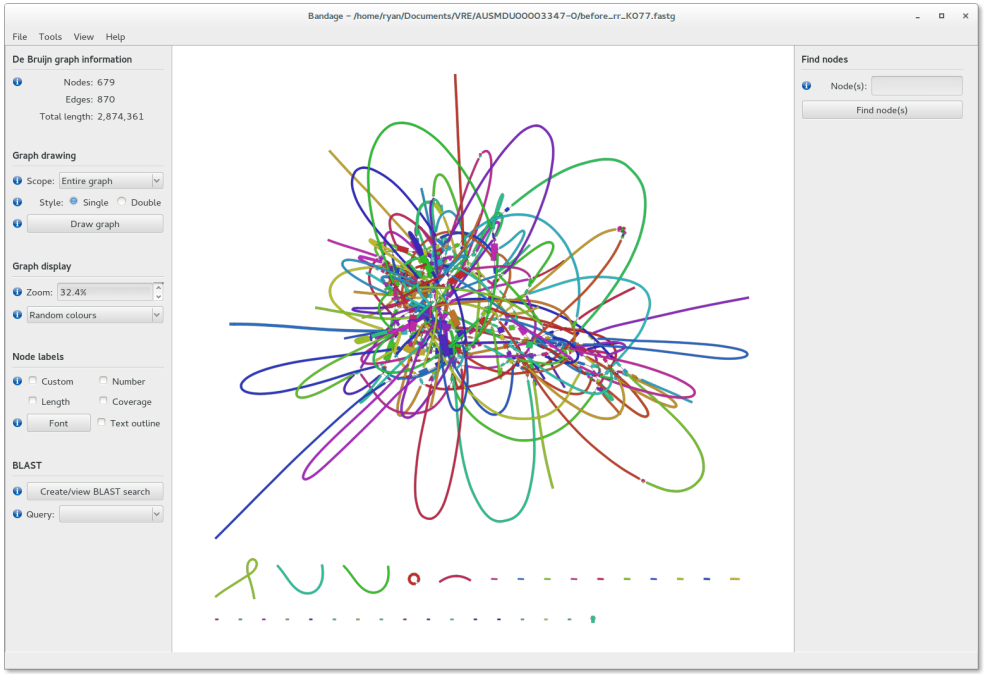
```
#!/bin/bash -l
#SBATCH --job-name spades
#SBATCH --output spades_out_%j.txt
#SBATCH --error spades_err_%j.txt
#SBATCH --time 2:00:00
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 1
#SBATCH --cpus-per-task 4
#SBATCH --mem 5000
#SBATCH --account project_2001379

module load biokit

spades.py -1 Matilda_R1_trimmed.fastq \
          -2 Matilda_R2_trimmed.fastq \
          -o SPADES_MATILDA \
          -t 4 \
          --isolate
```

SPAdes output

contigs.fasta:this contains the contigs generated by SPAdes

scaffolds.fasta: contigs generated by SPAdes after repeat resolution and scaffolding using paired-end information

assembly_graph.fastg: the assembly graph generated by SPAdes (can be viewed in Bandage)



Additional notes about Puhti

Batch jobs

`sbatch`

`scancel`

`squeue`

`seff`

```
Job ID: 3750729
Cluster: puhti
User/Group: stelmach/stelmach
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:15:57
CPU Efficiency: 74.53% of 00:21:24 core-walltime
Job Wall-clock time: 00:05:21
Memory Utilized: 2.49 GB
Memory Efficiency: 51.06% of 4.88 GB
Job consumed 0.40 CSC billing units based on following used resources
CPU BU: 0.36
Mem BU: 0.04
```

Assembly quality

What is a good assembly?

Good coverage throughout the contigs

Correct size (for bacteria, not 400 kb or 14 Mb)

As few contigs as possible

Similar GC across contigs

QUAST

http://bioinf.spbau.ru/quast

Tool to evaluate the quality of assemblies

After running QUAST, move to your computer with FileZilla:

- report.html
- report.pdf
- icarus.html
- icarus_viewers (folder)

Let’s assemble our genome

https://github.com/karkman/MMB-114_Genomics

(**Day 3:** Genome assembly)