Analyse de données NGS sous Galaxy

Migale facility

Valentin Loux - Cédric Midoux

2022-03-09

Practical informations

- 9h30 - 17h00
- 2 breaks morning and afternoon
- Lunch break of 1 hour
- Remote session … please be comprehensive !

Remote sessions rules

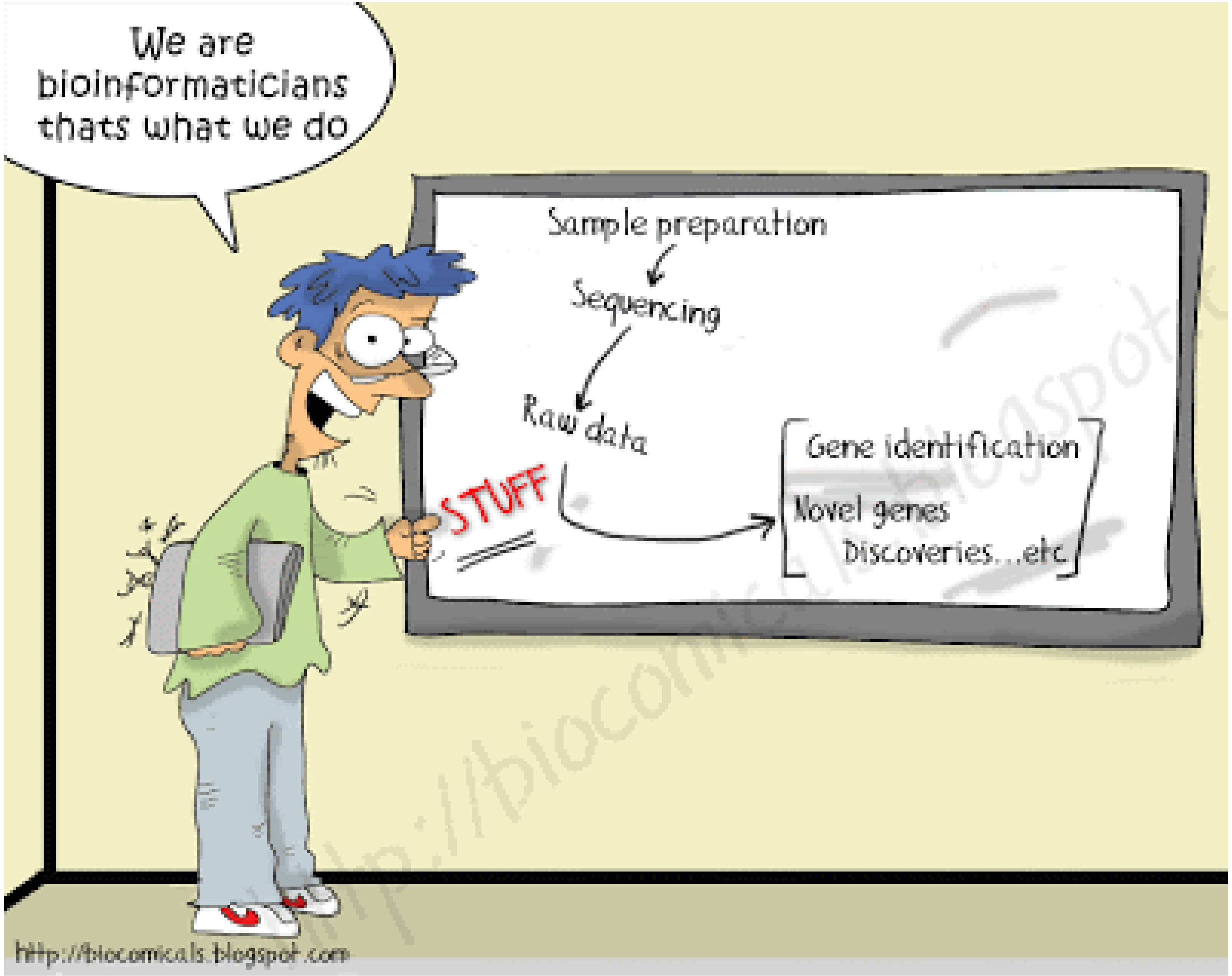
- Even remotely, it should be interactive !
- Please interrupt us :
  - raise (virtually) your hand
  - ask questions in the chat
- Practical session will be different :
  - tutorial support with practice to do on your own during a few minutes
  - group synchronization to be sure everyone follows
  - Inform us of your progress :
    - Ask (any) questions
    - Use green / red reactions

Ice breaking session

- Who are you?
  - Institution, laboratory, position …
- Why are your here
  - What are your needs in NGS data analysis?
- Do you have already dealt with NGS data?
  - Which kind of data?
  - Aim of the study?
- Have you ever used *Galaxy* ?

Migale team

- Migale website
- Infrastructure for bioinformatics
    - storage, compute
    - tools, databanks
    - interfaces (Galaxy)
- Dedicated service to Data Analysis
    - Specialists in Metagenomics
    - Bioinformatics & Statistics
    - More than 60 projects since 2016
    - Collaboration or Accompaniment


We are bioinformaticians thats what we do
Sample preparation
Sequencing
Raw data
STUFF
Gene identification
Novel genes
Discoveries...etc
http://biocomicals.blogspot.com

Objectives

After this training day, you will know:

- the characteristics of the main types of sequencers
- how to do a quality control of the raw sequences
- how to assemble a (small) genome
- how to align reads to a reference genome
- how to explore graphically an alignment
- how to compare assemblies

Program

Morning

- Introduction & Round table
- Sequencing technologies

Break

- Quality Control
- Data cleaning
- Assembly

Afternoon

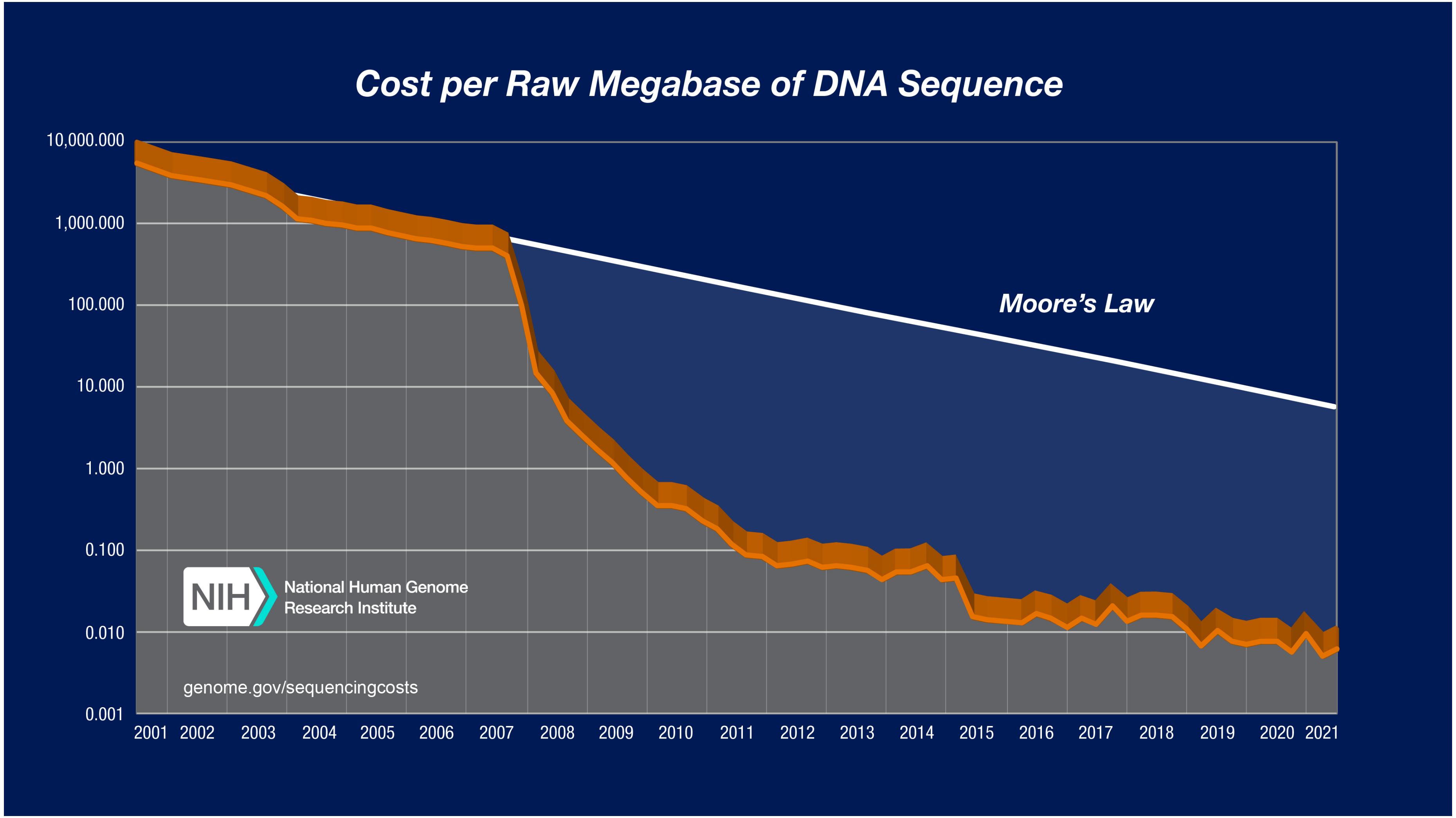
- Assembly evaluation and comparison

Break

- Mapping
- Visualisation

Next generation Sequencing in a few slides

Sequencing Cost per Megabase



Genome Sequencing, why ?

Interest in a genome that has not yet been sequenced

- Assembly and annotation
  - de novo sequencing
  - chromosomal rearrangements
  - metagenomics

A reference genome is available

- Alignment (mapping) of reads on the genome
  - Detection of genomic variants (SNPs)
  - RNA-seq (gene expression)
  - ChIP-seq (regulation of gene expression)
  - Chromosomal rearrangements, variation in gene copy number
  - Detection of small non-coding RNAs
  - metagenomics

Sequencing challenges

Smallest known (non viral) genome:

- *Carsonella ruddii* = 0.16 Mbp

Largest known genome:

- *Paris japonica* = 150 Gbp
- *Amoeba dubia* = 670 Gbp

Maximum Reads Size :

- 1st generation (Sanger): up to 900 bp
- 2nd generation: up to 500 bp
- 3rd generation: up to 100 - 1000 Kbp

Need to cut the genome into millions of fragments (**shotgun sequencing**) from the 2 DNA strands.

The operation to reconstruct the genetic elements from the raw reads is called **assembly**.

Sequencing technologies

- First generation :
    - Sanger sequencing
    - First step : fragment cloning
    - Reads up to 900 bp
    - Expensive
    - low throughput

Next generation Sequencing technologies

Second generation (since 2007)

- **454** - Sequencing by Synthesis - PCR Amplification
- **SOLiD** : Sequencing by Ligation - PCR Amplification
- **Ion Torrent** : Sequencing by Synthesis - PCR Amplification
- **Illumina** : Sequencing by Synthesis - PCR Amplification

Illumina : principles

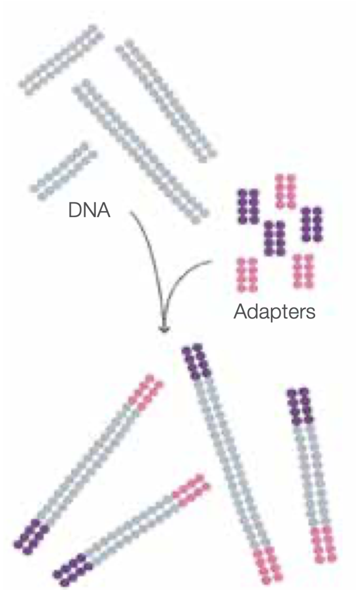
- Based on "reversible terminated chemistry" : reversible terminators that enable the identification of single nucleotides as they are washed over DNA strands.

Three steps :

- Amplification of DNA fragments
- Sequencing
- Analysis

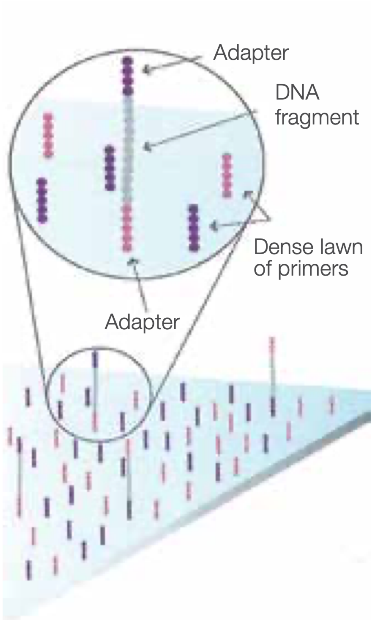
Reference : Technology Spotlight: Illumina® Sequencing

Prepare genomic DNA samples



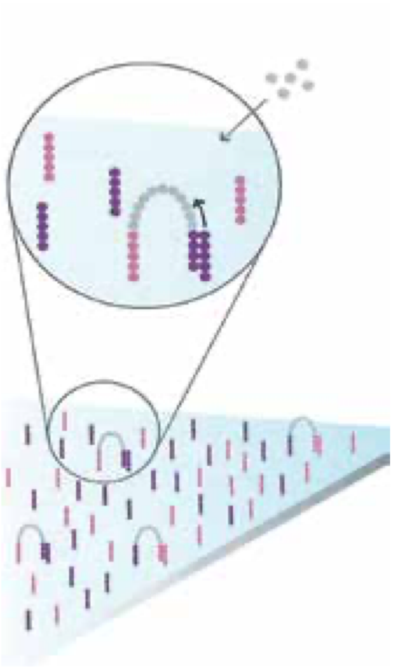
Randomly fragment genomic DNA and ligate adapters to both ends of the fragments

Attach DNA to Flow Cell Surface



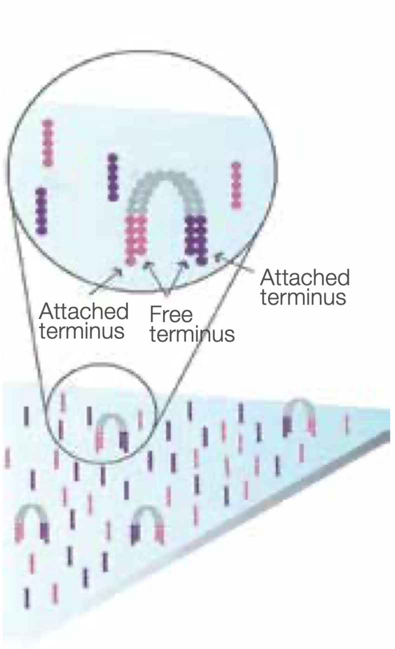
Bind single-stranded fragments randomly to the inside surface of the flow cell channels.

Bridge Amplification



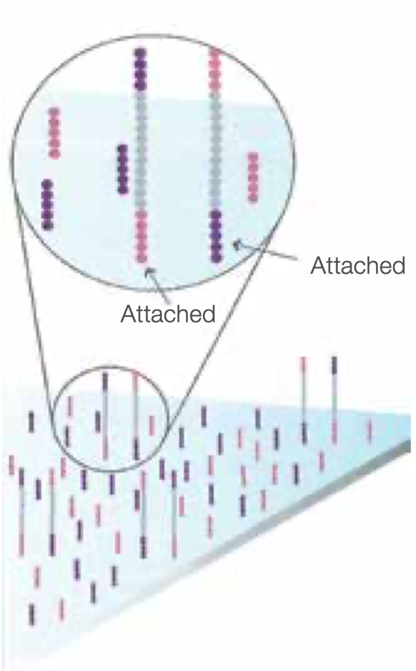
Add **unlabelled** nucleotides and enzyme to initiate solid-phase bridge amplification.

Fragments Become Double Stranded



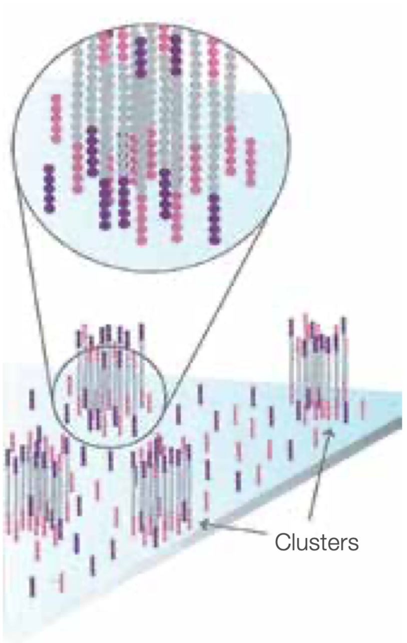
The enzyme incorporates nucleotides to build double-stranded bridges on the solid-phase substrate.

Denature the Double-Stranded Molecule



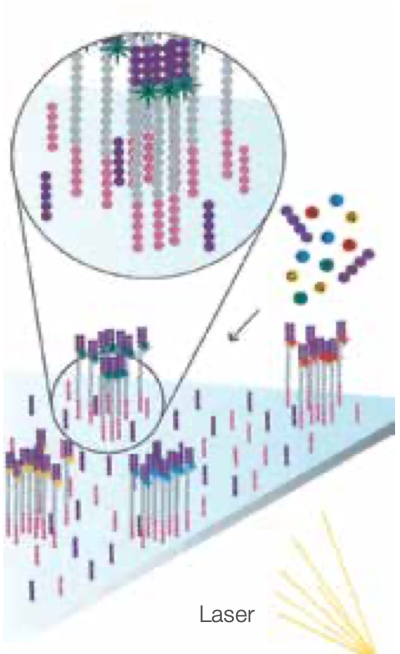
Denaturation leaves single-stranded templates anchored to the substrate.

Complete Amplification



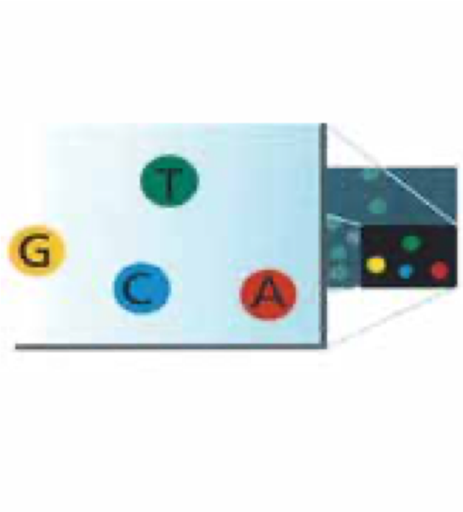
Several millions dense clusters of double-stranded DNA are grated in in channel of the flow cell.

Determine First Base



The first sequencing cycle begins by adding four labelled reversible terminators, primers, and DNA polymerase.

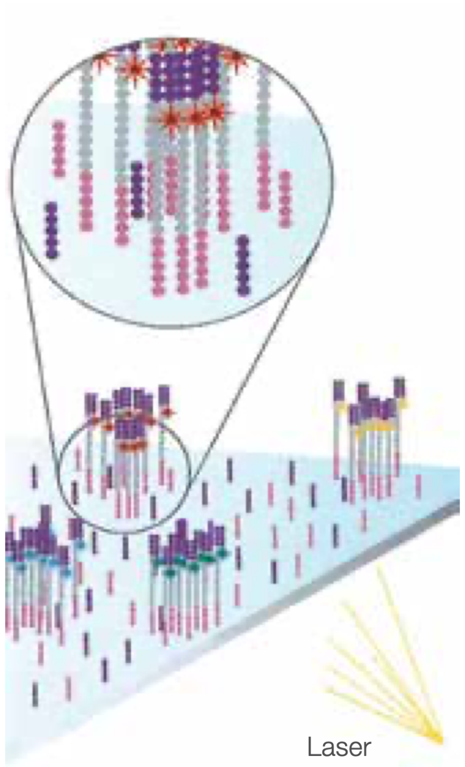
Image First Base



After laser excitation, the emitted fluorescence from each cluster is captured and the first base is identified.

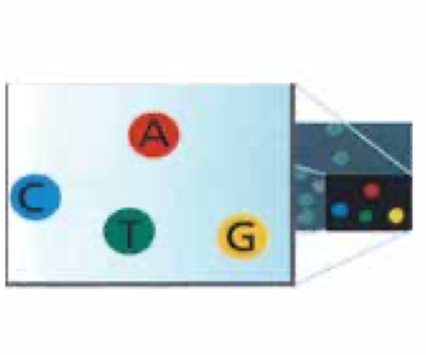
The blocked 3' terminus and florphore are removed,flow cell washed, leaving the terminator free for a second cycle.

Determine Second Base



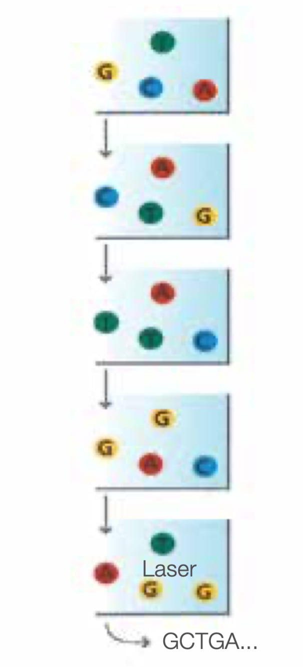
The next cycle repeats the incorporation of four labelled reversible terminators, primers, and DNA polymerase.

Image Second Chemistry Cycle



After laser excitation, the image is captured as before, and the identity of the second base is recorded.

Sequencing Over Multiple Chemistry Cycles



The sequencing cycles are repeated to determine the sequence of bases in a fragment, one base at a time.

Millions of clusters are processed in parallel, allowing high-throughput sequencing.

Illumina : summary

- High precision >99.5% (main type or errors : substitutions)
- Short reads (maximum 2 x 250)
- Huge throughput (up to 6 Tbp per run on NovaSeq)
- Some under-representation of rich AT- and GC- regions.
- Video

Sequencing - Glossary

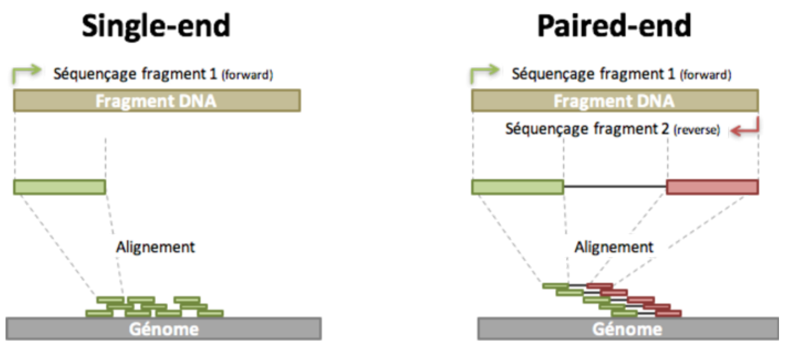
Read : piece of sequenced DNA

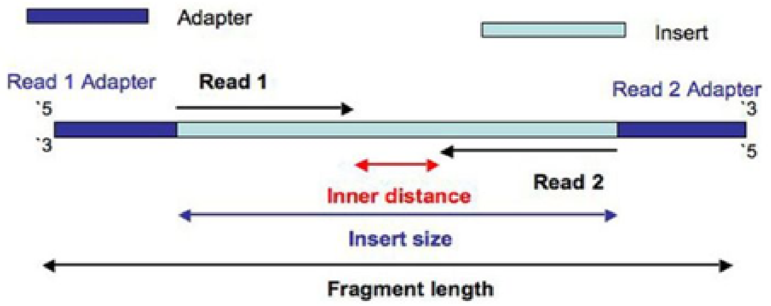
DNA fragment = 1 or more reads depending on whether the sequencing is single end or paired-end

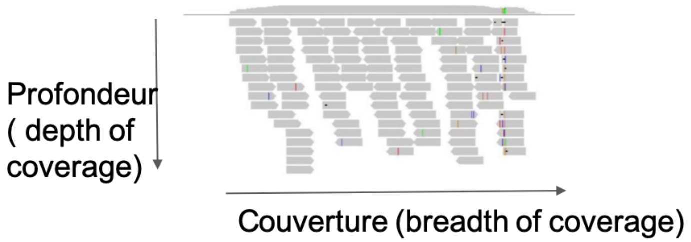
Insert = Fragment size

Depth = $N * L/G$
N = number of reads,
L = size,
G = genome size

Coverage = % of genome covered


Single-end
Séquençage fragment 1 (forward)
Fragment DNA
Alignement
Génome
Paired-end
Séquençage fragment 1 (forward)
Fragment DNA
Séquençage fragment 2 (reverse)
Alignement
Génome




Profondeur (depth of coverage)
Couverture (breadth of coverage)

3d generation

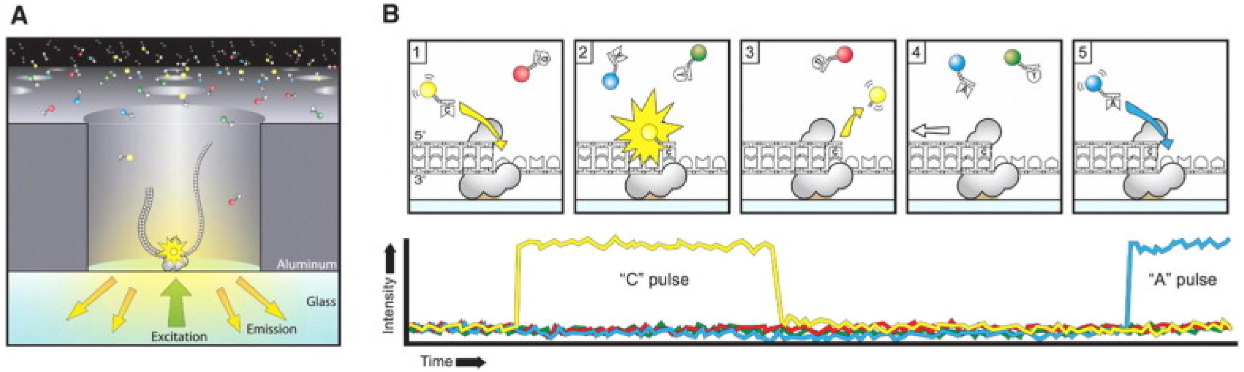
Target the weaknesses of the 2nd generation :

- PCR amplification
- Short reads

Two main competitors (in production) :

- Pacific Bioscience (PacBio)
- Oxford Nanopore Technologies (ONT)

PacBio



A polymerase is immobilized at the bottom of a sequencing unit called zero-mode waveguide (ZMW) .Four fluorescent-labelled nucleotides, which generate distinct emission spectrums, are added to the SMRT cell. As a base is held by the polymerase, a light pulse is produced that identifies the base. The replication processes in all ZMWs of a SMRT cell are recorded by a “movie” of light pulses, and the pulses corresponding to each ZMW can be interpreted to be a sequence of bases.

Reference

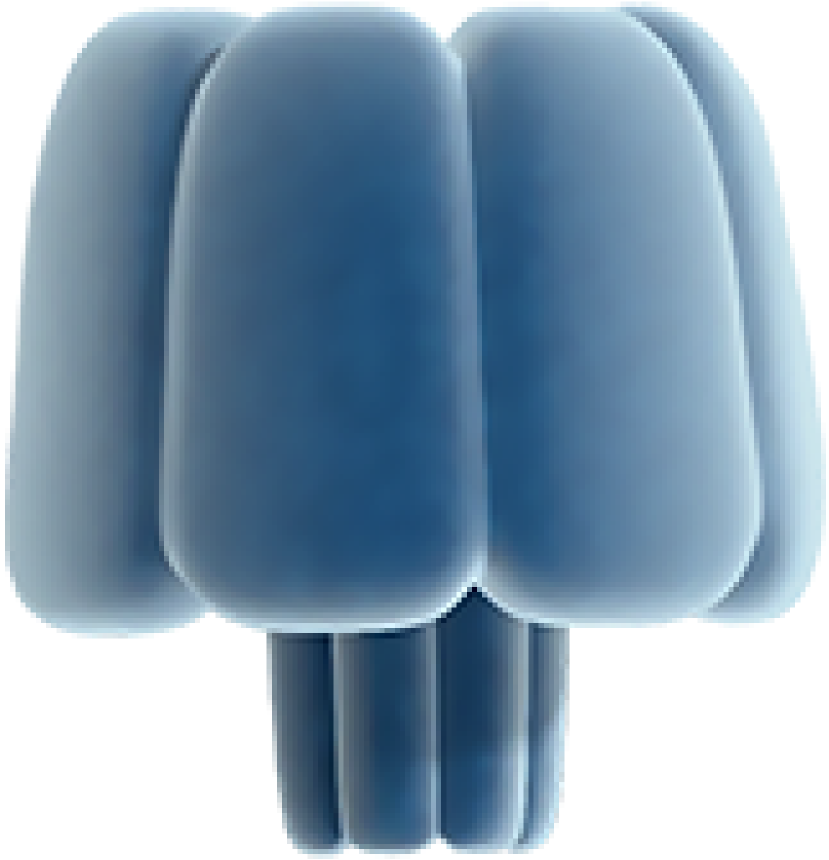
PacBio : summary

- Long reads (up to Kbs with SequelII)
- Depends on DNA quality
- High error rate. Tend to lower with depth
- Medium throughput

Applications :

- IsoSeq (RNA Isoform full length sequencing)
- Detection of DNA modification
- Assembly

Oxford Nanopore



Oxford Nanopore

MinION, GridION, PromethION

Sequencing on The ISS

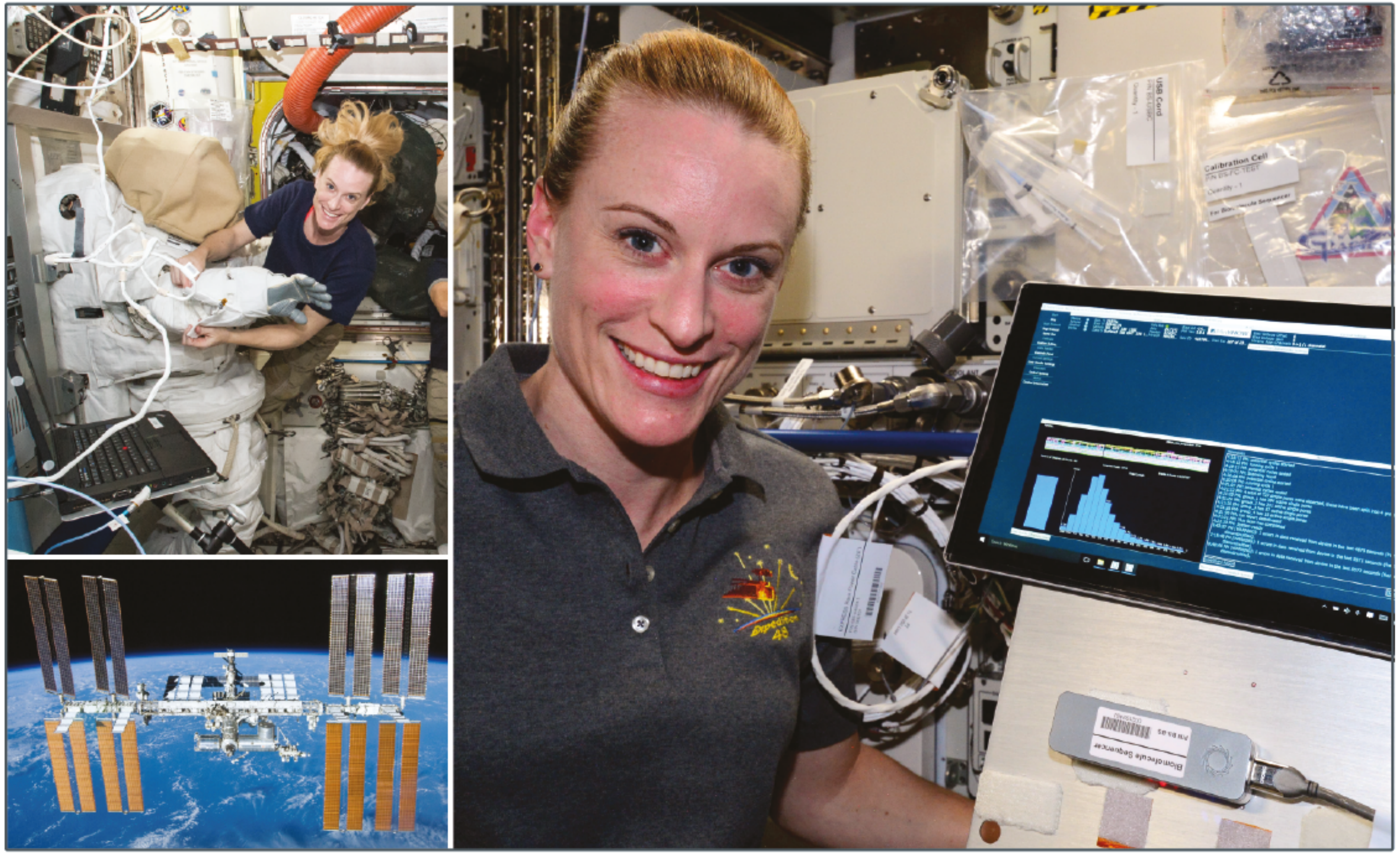


Fig. 1 Astronaut Kate Rubins on the ISS

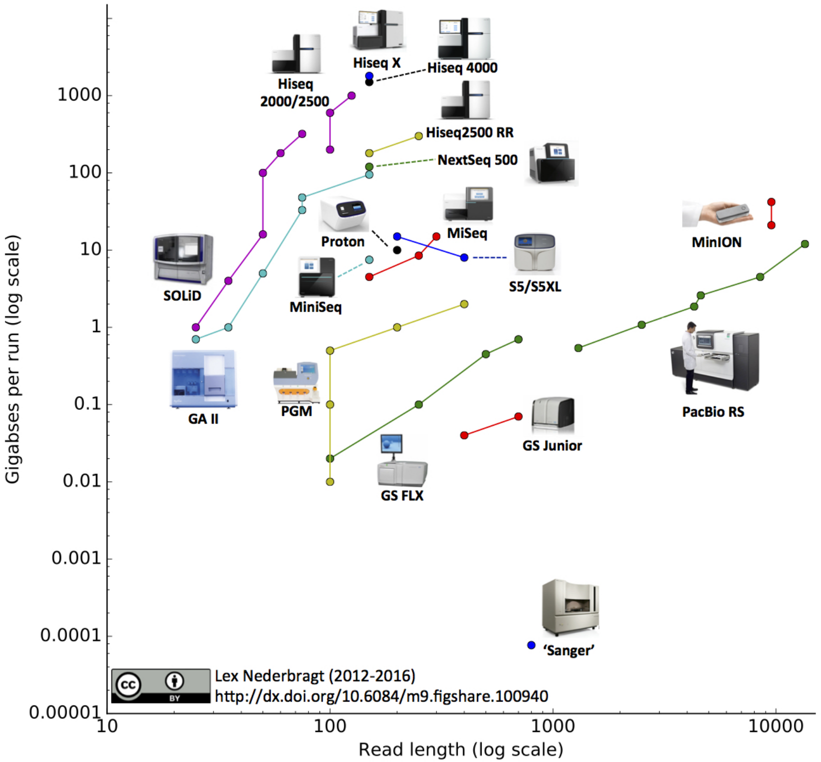
ONT Summary

- Ultra long reads (up to 1 Mb (!))
- Length of the reads depends on DNA quality
- Low to high throughput
- "On field" sequencing
- Direct RNA sequencing, peptide sequencing
- High error rate (5-10%), tends to lower with new chemistry, base calling algorithms and depth

Applications :

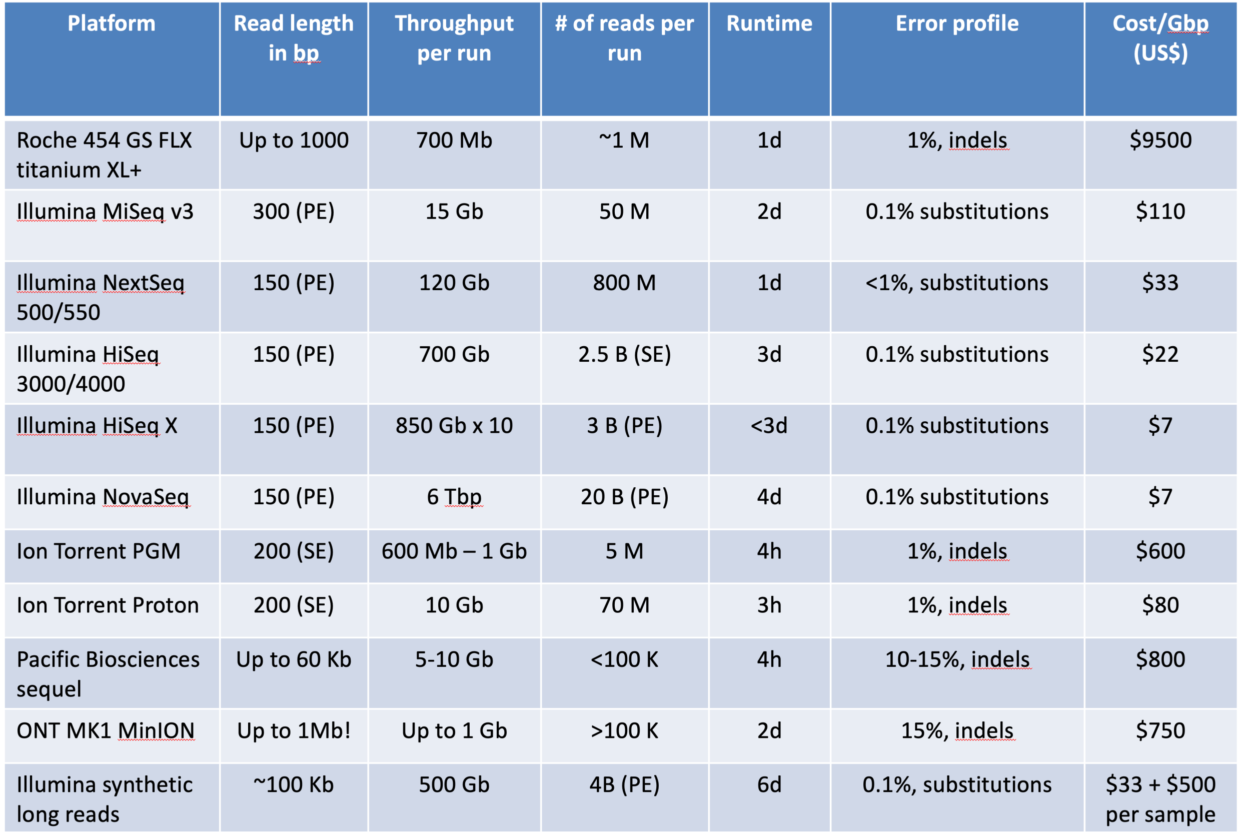
- Full length isofrom sequencing, direct RNA sequencing
- Detection of DNA modification
- Assembly

An other view on sequencing technologies (probably out of date)



Lex Nederbragt (2012-2016)
http://dx.doi.org/10.6084/m9.figshare.100940

Global Summary (probably out of date)



Platform	Read length in bp	Throughput per run	# of reads per run	Runtime	Error profile	Cost/Gbp (US$)
Roche 454 GS FLX titanium XL+	Up to 1000	700 Mb	~1 M	1d	1%, indels	$9500
Illumina MiSeq v3	300 (PE)	15 Gb	50 M	2d	0.1% substitutions	$110
Illumina NextSeq 500/550	150 (PE)	120 Gb	800 M	1d	<1%, substitutions	$33
Illumina HiSeq 3000/4000	150 (PE)	700 Gb	2.5 B (SE)	3d	0.1% substitutions	$22
Illumina HiSeq X	150 (PE)	850 Gb x 10	3 B (PE)	<3d	0.1% substitutions	$7
Illumina NovaSeq	150 (PE)	6 Tbp	20 B (PE)	4d	0.1% substitutions	$7
Ion Torrent PGM	200 (SE)	600 Mb – 1 Gb	5 M	4h	1%, indels	$600
Ion Torrent Proton	200 (SE)	10 Gb	70 M	3h	1%, indels	$80
Pacific Biosciences sequel	Up to 60 Kb	5-10 Gb	<100 K	4h	10-15%, indels	$800
ONT MK1 MinION	Up to 1Mb!	Up to 1 Gb	>100 K	2d	15%, indels	$750
Illumina synthetic long reads	~100 Kb	500 Gb	4B (PE)	6d	0.1%, substitutions	$33 + $500 per sample

An interesting review (Goodwin, McPherson, and McCombie, 2016)

Nature review : Milestones in Genomic Sequencing

Switch to Hands-on :

Connect to Galaxy

Practical session :

- *Escherichia coli* genome (re)sequencing
    - Illumina MiSeq
    - Paired-end sequencing (2*150bp , insert size ~300bp)
    - Sub-sampled

Connect to Galaxy

- https://galaxy.migale.inrae.fr
- Login : **stageXX**
- Data in `Shared Data / Data Libraries / formation NGS / Reads`
- References in `Shared Data / Data Libraries / formation NGS / Refs`

FASTQ format

FASTQ syntax

The FASTQ format is the de facto standard by which all sequencing instruments represent data. It may be thought of as a variant of the FASTA format that allows it to associate a quality measure to each sequence base: **FASTA with QUALITIES**.

FASTQ syntax

The FASTQ format consists of 4 sections:

1. A FASTA-like header, but instead of the `>` symbol it uses the `@` symbol. This is followed by an ID and more optional text, similar to the FASTA headers.
2. The second section contains the measured sequence (typically on a single line), but it may be wrapped until the `+` sign starts the next section.
3. The third section is marked by the `+` sign and may be optionally followed by the same sequence id and header as the first section
4. The last line encodes the quality values for the sequence in section 2, and must be of the same length as section 2.

FASTQ syntax

Example

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

FASTQ quality

Each character represents a numerical value: a so-called Phred score, encoded via a single letter encoding.

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
|    |    |    |    |    |    |    |    |
0....5...10...15...20...25...30...35...40
|    |    |    |    |    |    |    |    |
worst................................best
```

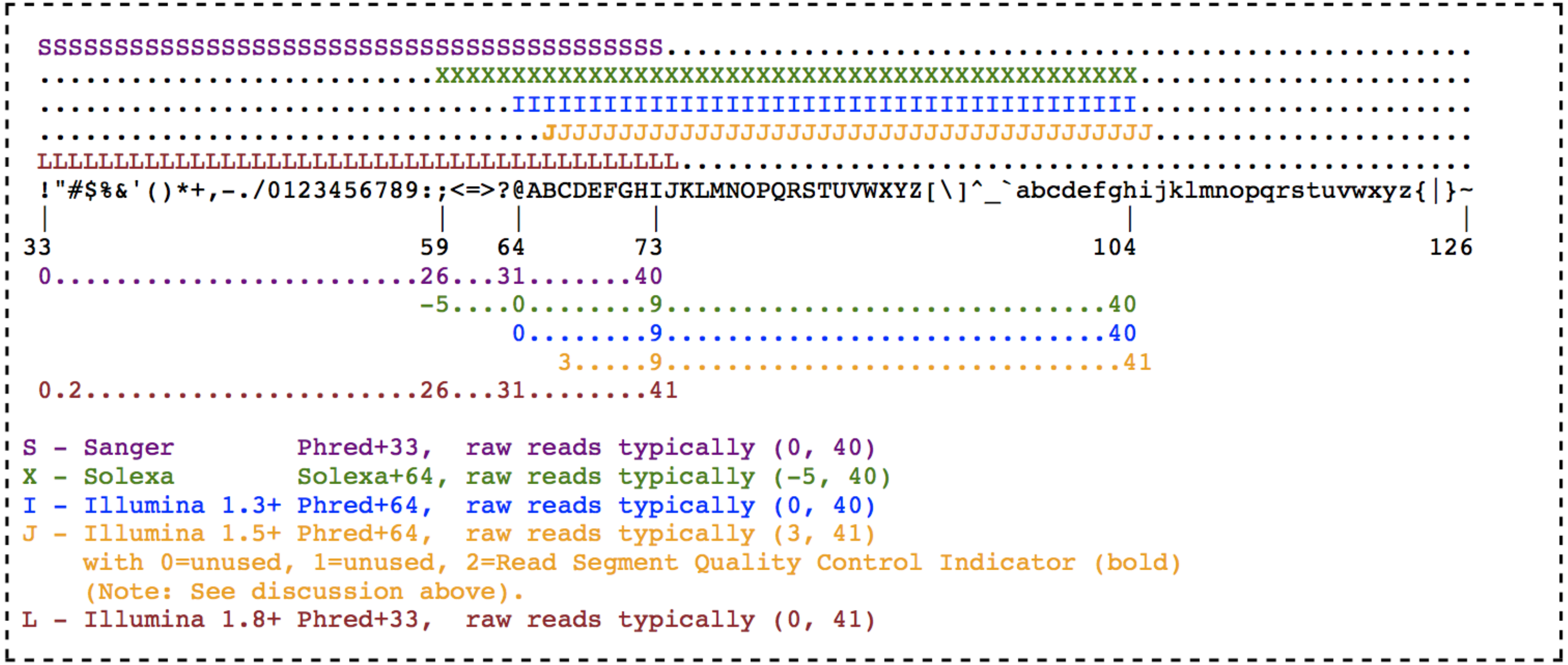
The numbers represent the error probabilities via the formula: $Error = 10^{-P/10}$

It is basically summarized as:

- P=0 means 1/1 (100% probability of error)
- P=10 means 1/10 (10% probability of error)
- P=20 means 1/100 (1% probability of error)
- P=30 means 1/1000 (0.1% probability of error)
- P=40 means 1/10000 (0.01% probability of error)

FASTQ quality encoding specificities

There was a time when instrumentation makers could not decide at what character to start the scale. The **current standard** shown above is the so-called Sanger (+33) format where the ASCII codes are shifted by 33. There is the so-called +64 format that starts close to where the other scale ends.



```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
  ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
  .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
  LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                         |    |        |                              |                     |
 33                        59   64       73                            104                   126
  0........................26...31.......40
                           -5....0........9..............................40
                                 0........9..............................40
                                    3.....9...............................41
  0.2.......................26...31........41

 S - Sanger        Phred+33,  raw reads typically (0, 40)
 X - Solexa        Solexa+64, raw reads typically (-5, 40)
 I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
 J - Illumina 1.5+ Phred+64,  raw reads typically (3, 41)
     with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
     (Note: See discussion above).
 L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

FASTQ Header informations

Information is often encoded in the “free” text section of a FASTQ file.

`@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG` contains the following information:

- `EAS139`: the unique instrument name
- `136`: the run id
- `FC706VJ`: the flowcell id
- `2`: flowcell lane
- `2104`: tile number within the flowcell lane
- `15343`: ‘x’-coordinate of the cluster within the tile
- `197393`: ‘y’-coordinate of the cluster within the tile
- `1`: the member of a pair, 1 or 2 (paired-end or mate-pair reads only)
- `Y`: Y if the read is filtered, N otherwise
- `18`: 0 when none of the control bits are on, otherwise it is an even number
- `ATCACG`: index sequence

This information is specific to a particular instrument/vendor and may change with different versions or releases of that instrument.

Switch to Hands-on :

Fastq import & visualisation

Quality control

Why QC'ing your reads ?

What are the information you want to know about the sequencing when you perform Quality Control ?

Collective Answer on this collaborative whiteboard

Why QC'ing your reads ?

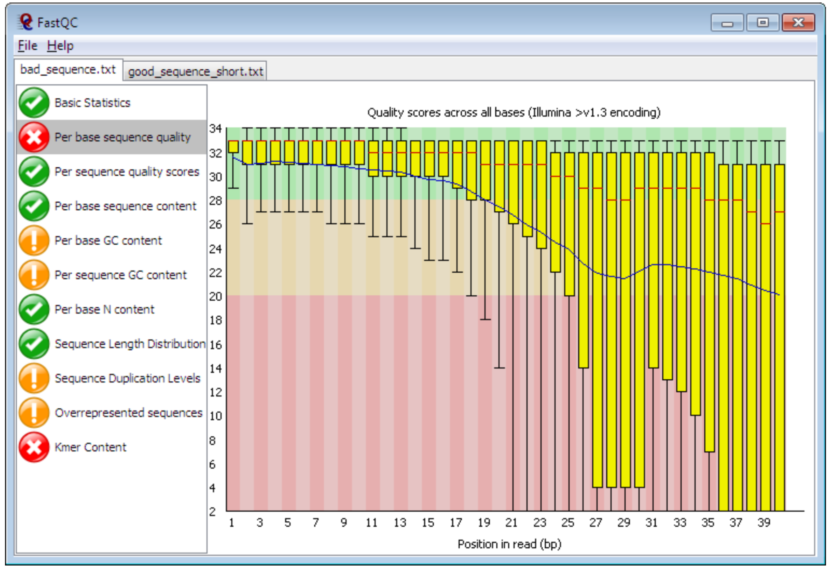
Try to answer to (not always) simple questions:

- Are data conform to the expected level of performance?
  - Size
  - Number of reads
  - Quality
- Residual presence of adapters or indexes ?
- Are there (un)expected technical biases
- Are there (un)expected biological biases

> Quality control without context leads to misinterpretation

Quality control for FASTQ files

- FastQC (Andrews, 2010)
  - QC for (Illumina) FastQ files
  - Command line fastqc or graphical interface
  - Complete HTML report to spot problem originating from sequencer, library preparation, contamination
  - Summary graphs and tables to quickly assess your data



- ✓ Basic Statistics
- ✗ Per base sequence quality
- ✓ Per sequence quality scores
- ✓ Per base sequence content
- ! Per base GC content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ! Sequence Duplication Levels
- ! Overrepresented sequences
- ✗ Kmer Content

- https://rtsf.natsci.msu.edu/genomics/tech-notes/fastqc-tutorial-and-faq/

Switch to Hands-on :

Quality Control with FastQC

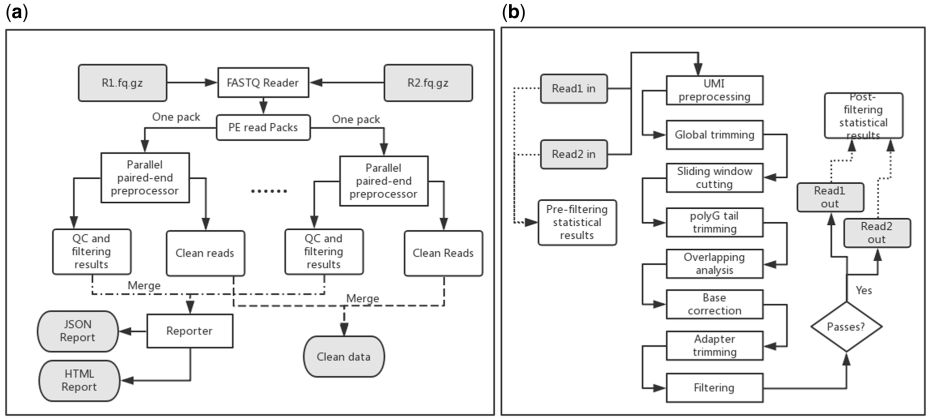
Reads cleaning

Objectives

- Detect and remove sequencing adapters (still) present in the FastQ files
- Filter / trim reads according to quality (as plotted in FastQC)

Tools

- Simple & fast : Sickle (Joshi and Fass, 2011) (quality), cutadapt (Martin, 2011) (adpater removal)
- Ultra-configurable : Trimmomatic
- All in one & ultra-fast : fastp (Zhou, Chen, Chen, and Gu, 2018)



main workflow

paired-end preprocessor

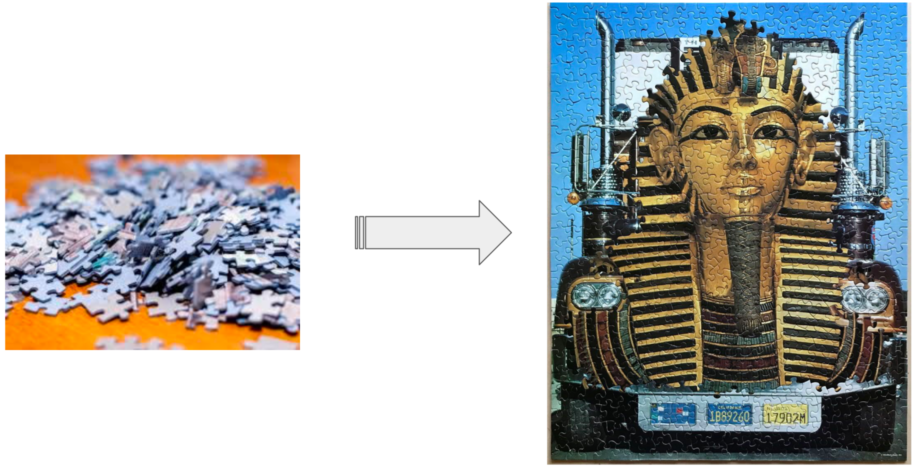
Switch to hands-on :

Clean your data with Sickle

Assembly : principles

Similar to a puzzle :

- millions of pieces -without the original image
- with pieces in both sense
- the pieces do not necessarily fit together (sequencing errors)
- parts of the puzzle are missing (cover + sequencing bias)



Assembly

All assembly algorithms are based on read overlap.

- Different ways of calculating overlap :
- "All vs All" comparison :
  - "old" assemblers based on this approach
  - Graph representing overlap between reads
  - Quadratic number of comparison (number of reads^2)
  - do not scale with billion of reads
- de Bruijn Graph
  - Named after Nicolaas Govert de Bruijn
  - Directed graph representing overlaps between sequences of symbols
  - Sequences can be reconstructed by moving between nodes in graph

Slide Credits

De Bruijn Graph

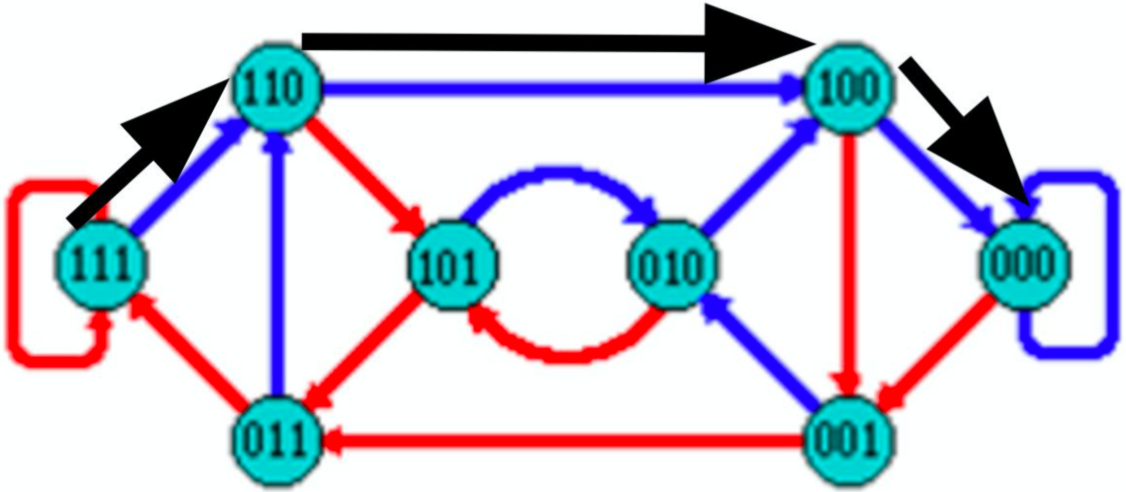
- A directed graph of sequences of symbols
- Nodes in the graph are k-mers
- Edges represent consecutive k-mers (which overlap by k-1 symbols)

Consider the 2 symbol alphabet (0 & 1) de Bruijn Graph for k =3

Producing sequences

- Sequences of symbols are produced by moving through the graph
  e.g. 111000 = 111 -> 110 -> 100 -> 000

```
1 1 1
  1 1 0
    1 0 0
      0 0 0
-----------
1 1 1 0 0 0
```

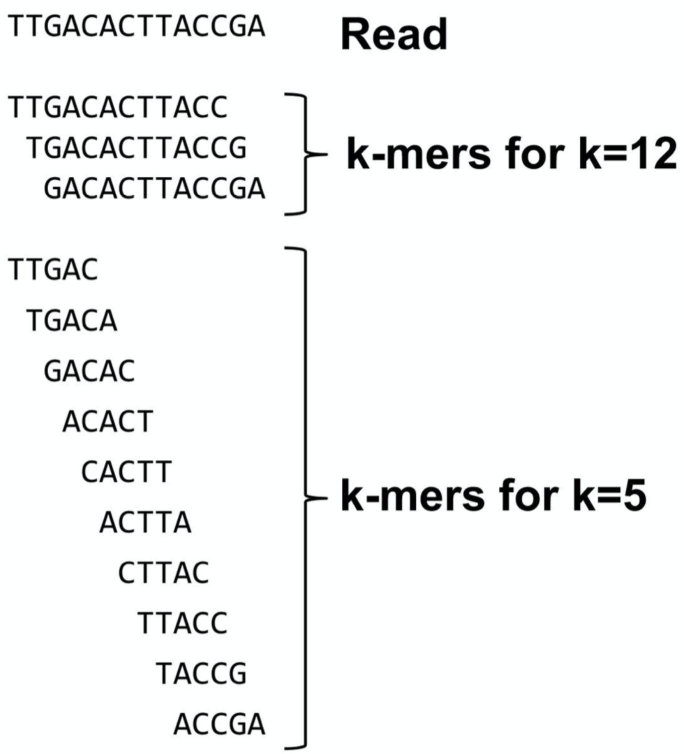


K-mers ?

- To be able to use de Bruijn graphs, we need reads of **length L to overlap by L-1 bases**.
- Not all reads will overlap another read perfectly.
  - Read errors
  - Coverage "holes"
- Not all reads are the same length (depending on technology and quality clean-up)

To help us get around these problems, we use all k-length subsequences of the reads, these are the k-mers.

What are K-mers ?



TTGACACTTACCGA

Read

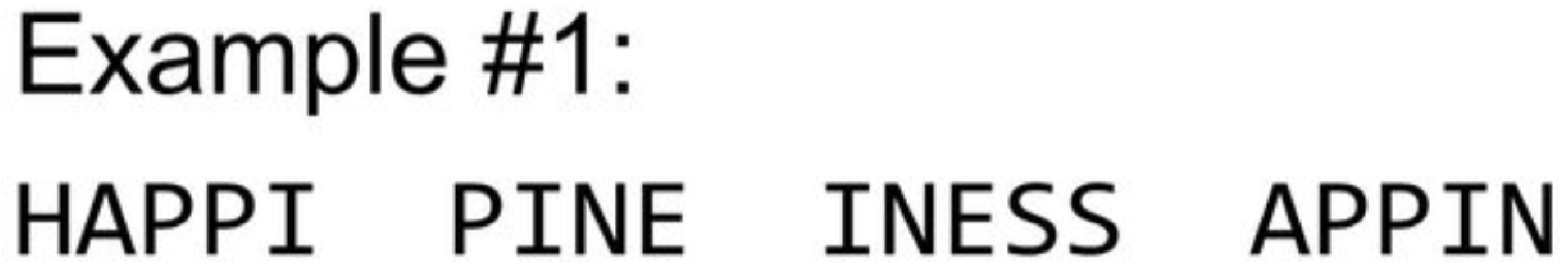
TTGACACTTACC
TGACACTTACCG
GACACTTACCGA

k-mers for k=12

TTGAC
TGACA
GACAC
ACACT
CACTT
ACTTA
CTTAC
TTACC
TACCG
ACCGA

k-mers for k=5

K-mers de Bruijn graph



Example #1:

HAPPI   PINE   INESS   APPIN

K-mers de Bruijn graph

K-mers de Bruijn graph

Example #1:

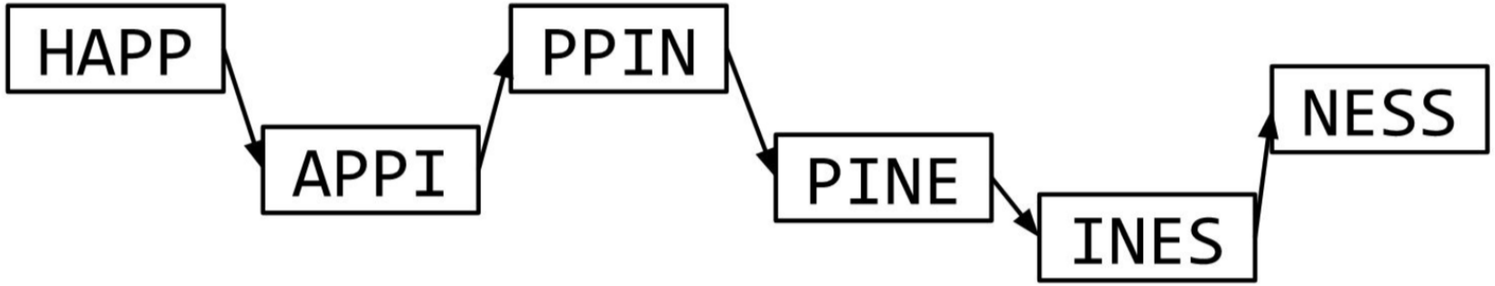
HAPPI PINE INESS APPIN

k = 4 k-mers:

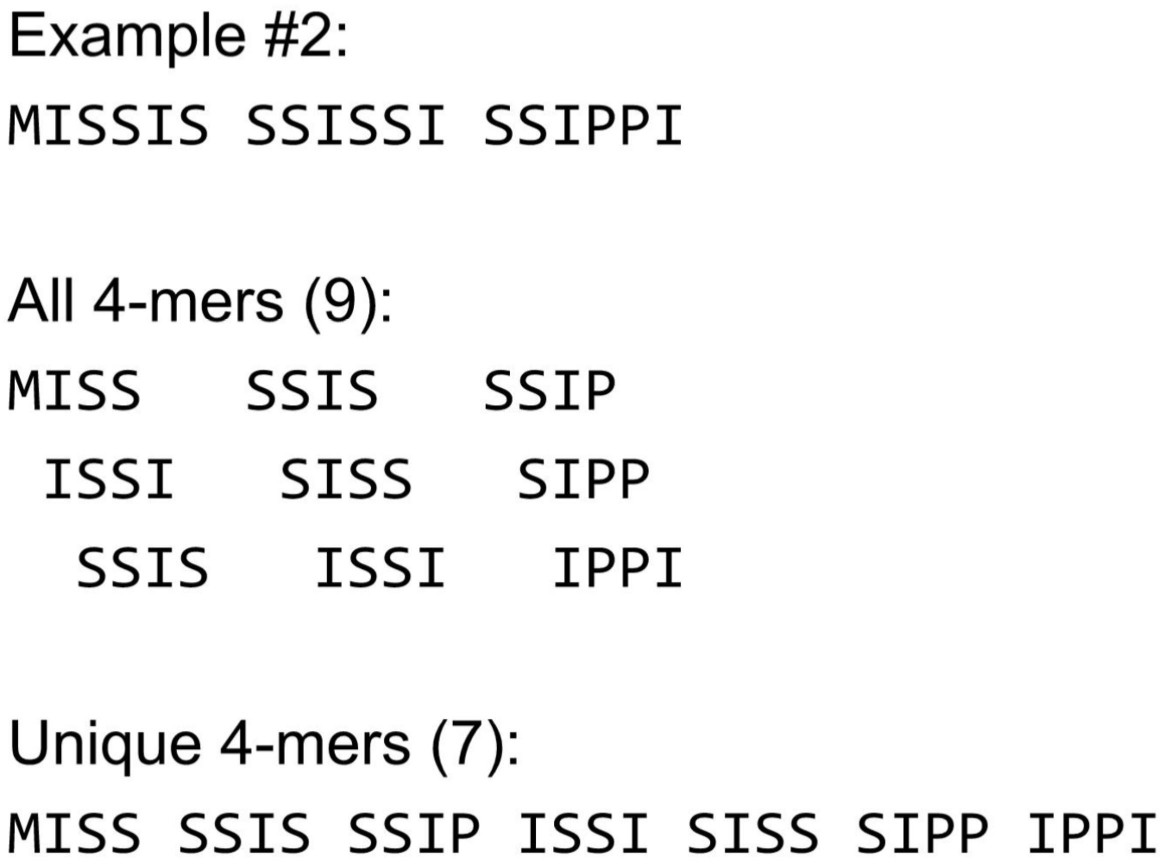
HAPP APPI

PINE PPIN

INES NESS



The problem of repeats



Example #2:

```
MISSIS SSISSI SSIPPI
```

All 4-mers (9):

```
MISS   SSIS   SSIP
 ISSI   SISS   SIPP
  SSIS   ISSI   IPPI
```

Unique 4-mers (7):

```
MISS SSIS SSIP ISSI SISS SIPP IPPI
```

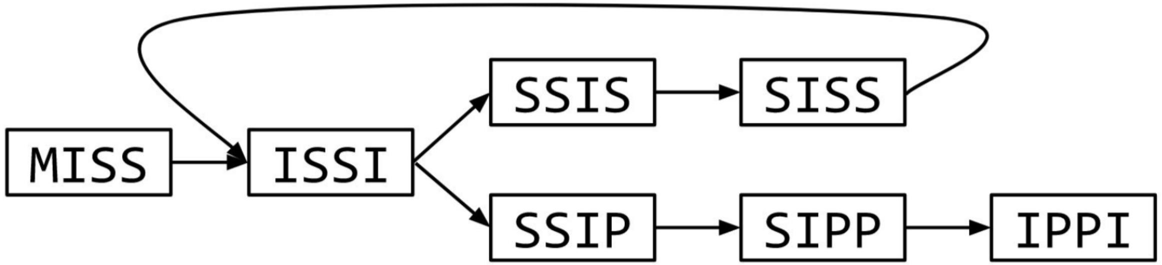
The problem of repeats

Example #2:

MISSIS SSISSI SSIPPI

All 4-mers:

MISS ISSI SSIS SISS SSIP SIPP IPPI



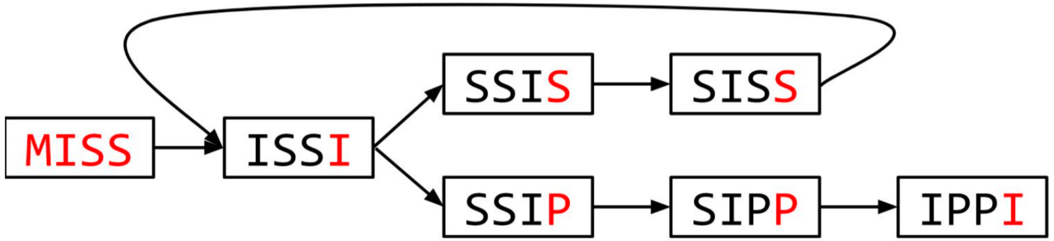
The problem of repeats

Example #2:

MISSIS SSISSI SSIPPI

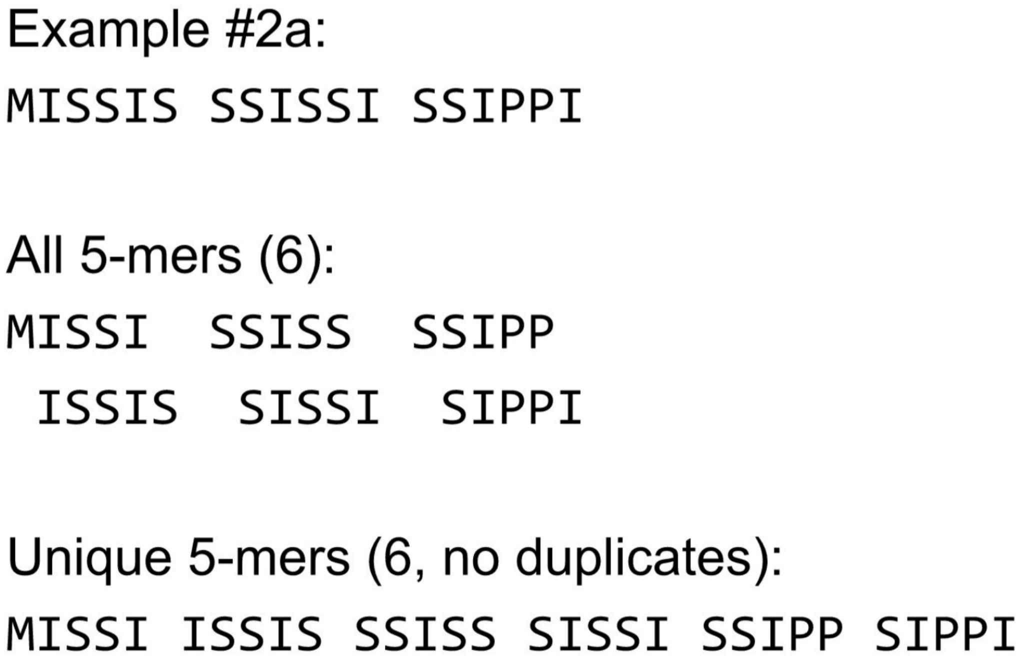
All 4-mers:

MISS ISSI SSIS SISS SSIP SIPP IPPI



MISSISSIPPI or MISSISSISSISSIPPI or ...

Different k



Example #2a:

```
MISSIS SSISSI SSIPPI
```

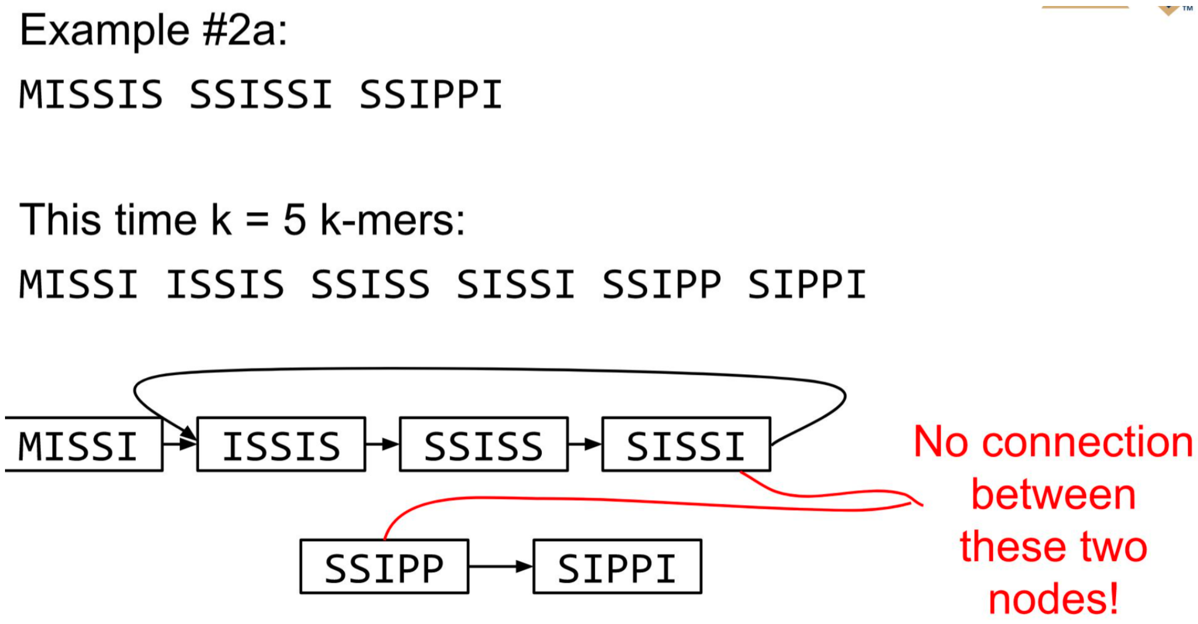
All 5-mers (6):

```
MISSI  SSISS  SSIPP
 ISSIS  SISSI  SIPPI
```

Unique 5-mers (6, no duplicates):

```
MISSI ISSIS SSISS SISSI SSIPP SIPPI
```

Different k



Example #2a:

MISSIS SSISSI SSIPPI

This time k = 5 k-mers:

MISSI ISSIS SSISS SISSI SSIPP SIPPI

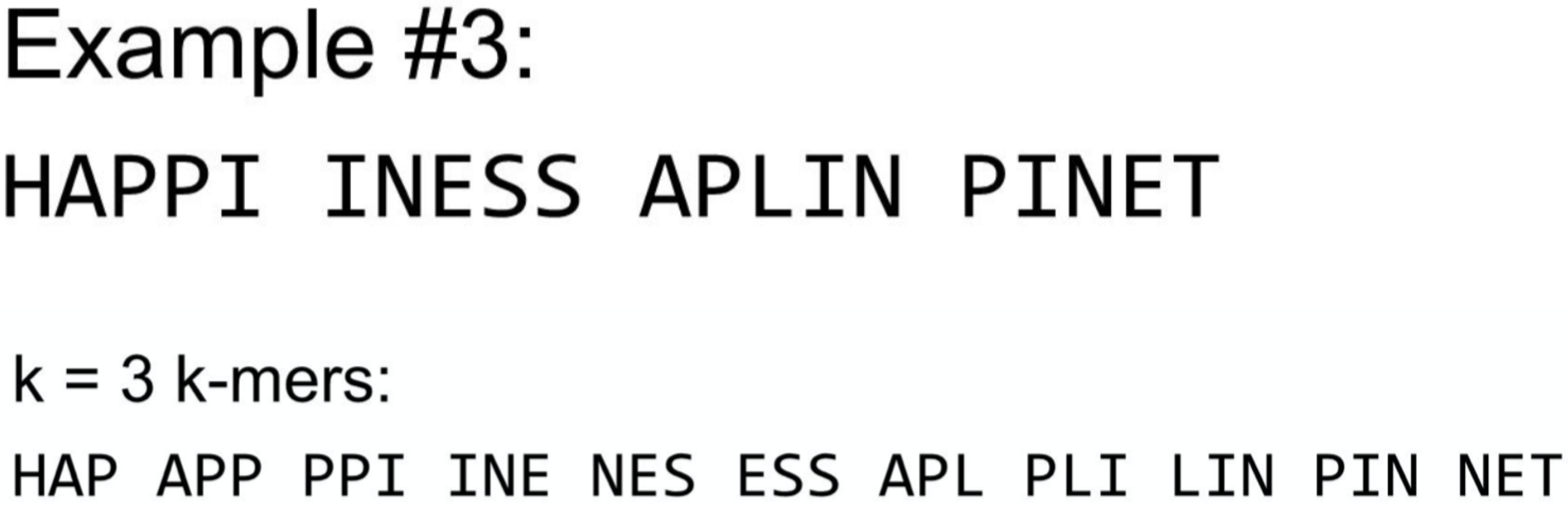
2 contigs : *MISSISSIS SSIPPI*

Choose k wisely

- Lower k
    - More connections
    - Less chance of resolving small repeats
    - Higher k-mer coverage
- Higher k
    - Less connections
    - More chance of resolving small repeats
    - Lower k-mer coverage

Optimum value for k will balance these effects.

Sequencing errors



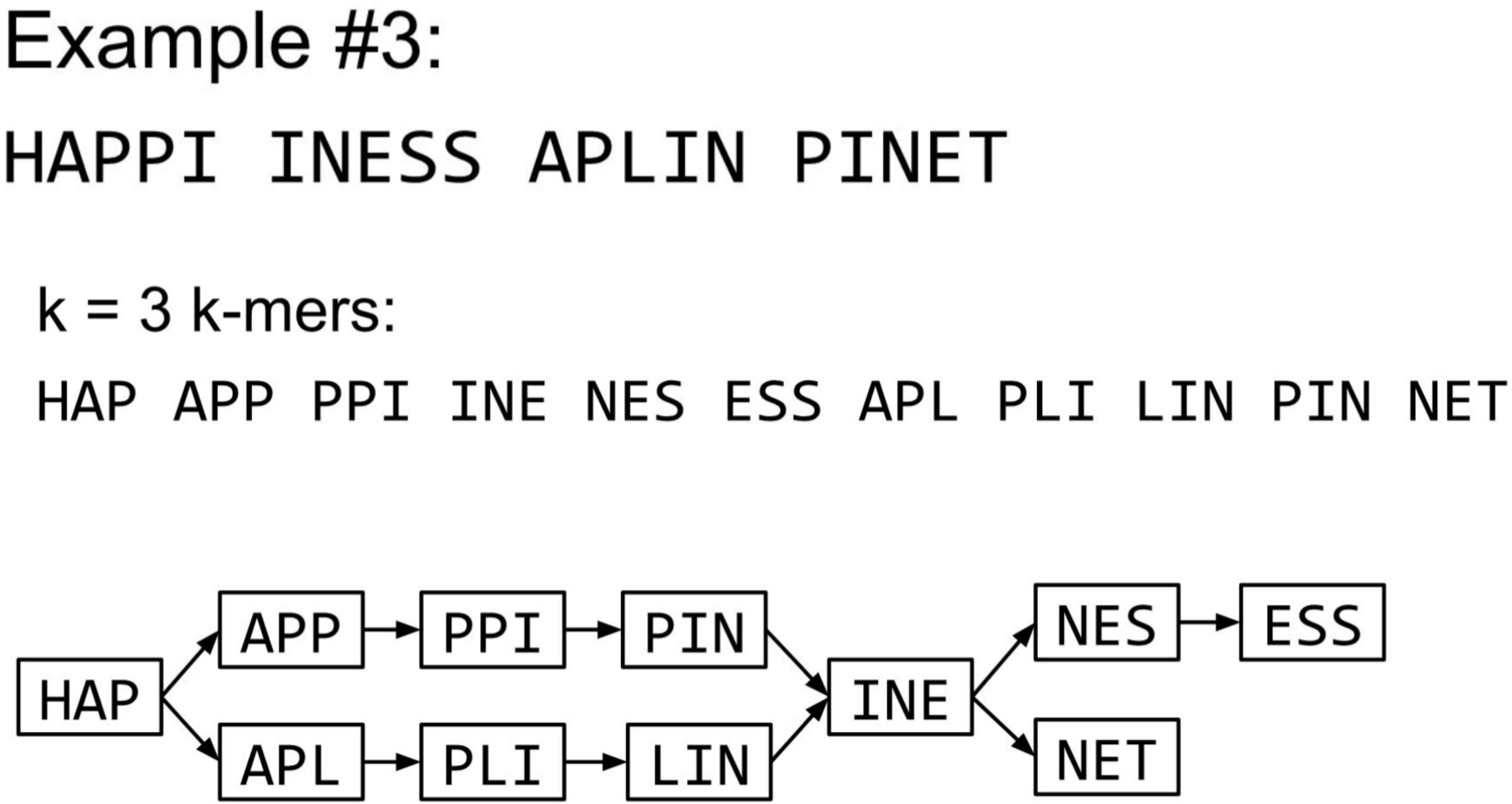
Example #3:

HAPPI INESS APLIN PINET

k = 3 k-mers:

HAP APP PPI INE NES ESS APL PLI LIN PIN NET

Sequencing errors



Example #3:

HAPPI INESS APLIN PINET

k = 3 k-mers:

HAP APP PPI INE NES ESS APL PLI LIN PIN NET

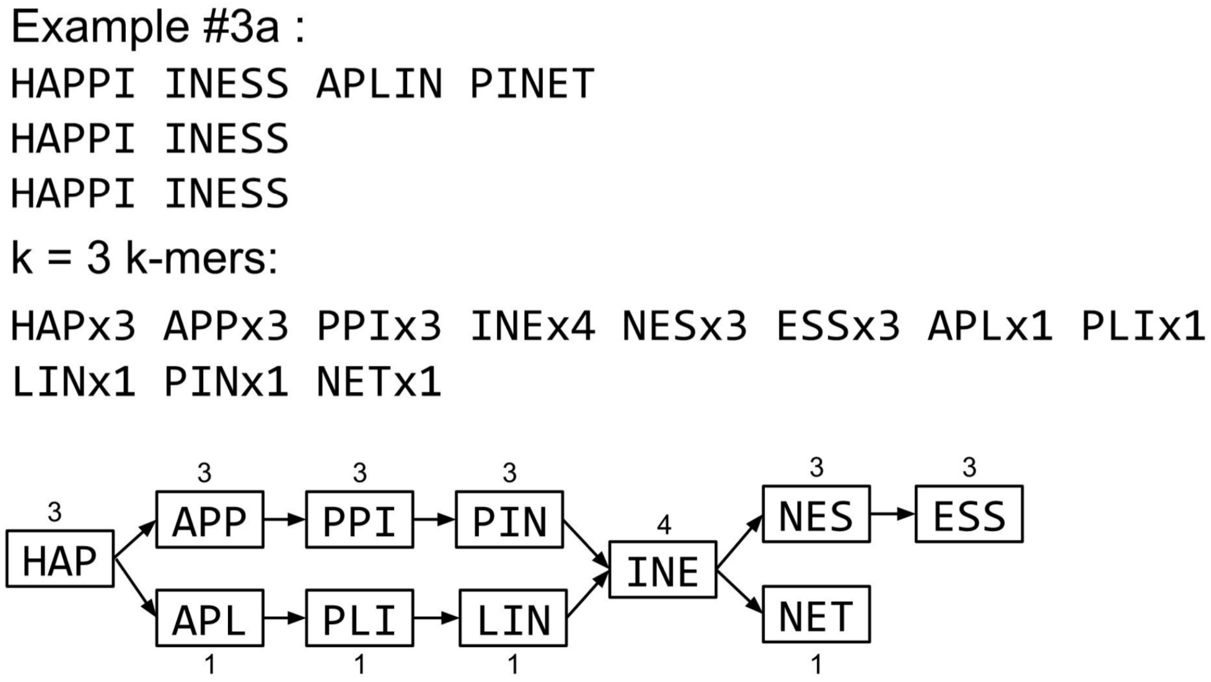
Sequencing errors

More coverage

- Errors won't be duplicated in every read
- Most reads will be error free
- We can count the frequency of each k-mer
- Annotate the graph with the frequencies
- Use the frequency data to clean the de Bruijn graph

More coverage depth will help overcome errors!

Sequencing errors - coverage



Example #3a :

```
HAPPI INESS APLIN PINET
HAPPI INESS
HAPPI INESS
```

k = 3 k-mers:

```
HAPx3 APPx3 PPIx3 INEx4 NESx3 ESSx3 APLx1 PLIx1
LINx1 PINx1 NETx1
```

Which path looks most valid ? Why ?

An important parameter : coverage cutoff

- At what point is a low coverage indicative of an error?
- Can we ignore low coverage nodes and paths?
- This is a new assembly parameter

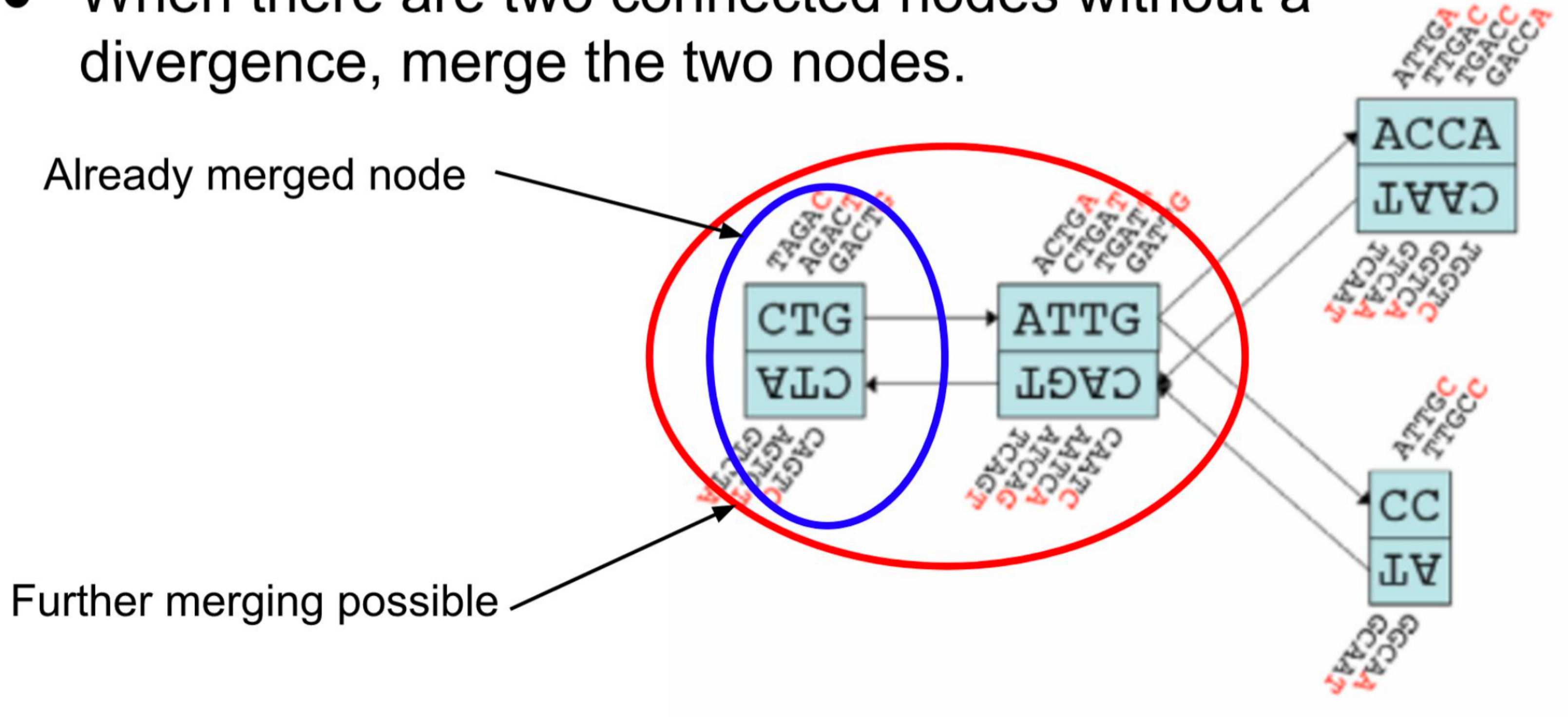
Coverage cut-off is an important parameter to differentiate error from real variations

de Bruijn Graph Assembly process

1. Select a value for k
2. "Hash" the reads (make the kmers)
3. Count the kmers
4. Make the de Bruijn graph
5. **Perform graph simplification steps** - use cov cutoff
6. Read off contigs from simplified graph

Graph simplification : Chain Merging

- When there are two connected nodes without a divergence, merge the two nodes.



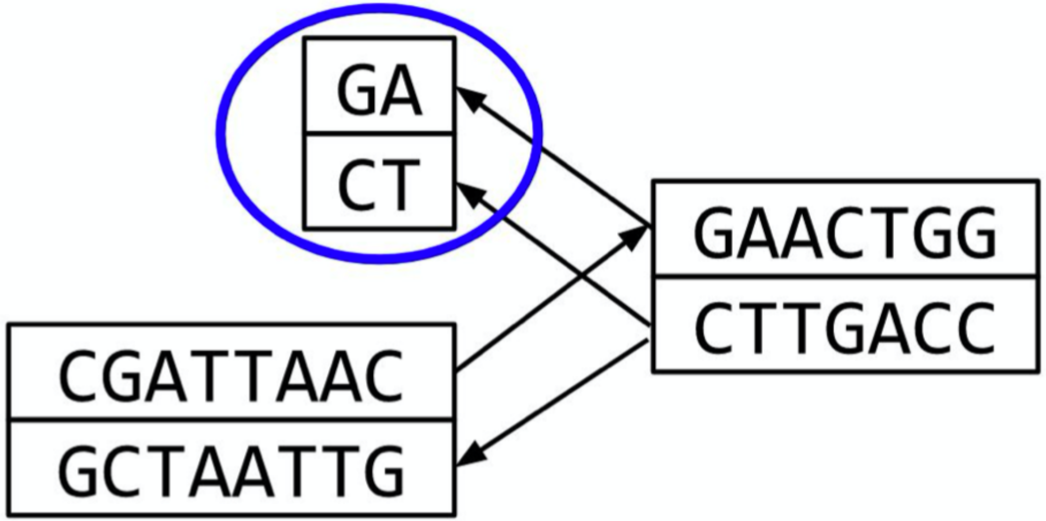
Already merged node

Further merging possible

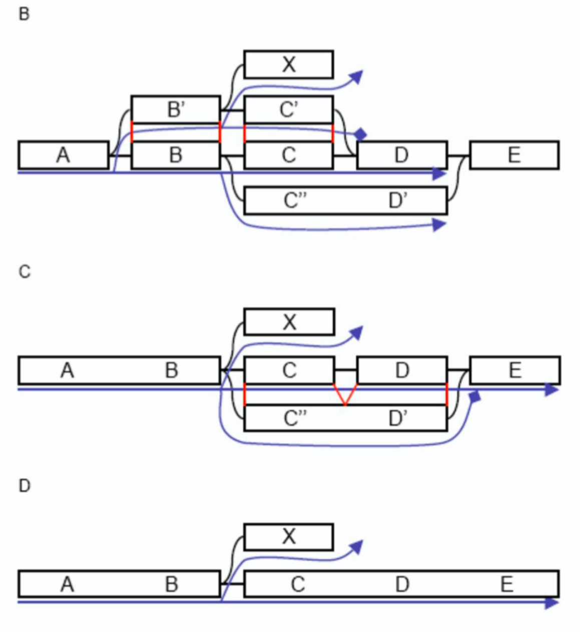
Zerbino & Birney, 2008

Graph simplification : Tip Clipping

- Clip tips if the length of the tip is < 2 x k



Graph simplification : Bubble Collapsing



Step 3: Bubble collapsing

- Detect redundant paths through graph
- Compare the paths using sequence alignment
- If similar, merge the paths

Image: Zerbino & Birney 2008

Make contigs

- Find an unbalanced node in the graph
- Follow the chain of nodes and "read off" the bases to produce the contigs
- Where there is an ambiguous divergence/convergence, stop the current contig and start a new one.
- Re-trace the reads through the contigs to help with repeat resolution

Graph simplification : Remove low coverage nodes

- remove erroneous nodes and edges using the "**coverage cutoff**"
- genuine short nodes will be kept because of their high coverage

Assemble with SPADES

SPADES (Bankevich, Nurk, Antipov, Gurevich, Dvorkin, Kulikov, Lesin, Nikolenko, Pham, Prjibelski, and others, 2012)is the de Bruijn graph assembler by Pavel Pevzner's group out of St. Petersburg

- Uses multiple k-mers to build the graph
  - Graph has connectivity and specificity
  - Usually use a low, medium and high k-mer size together.
- Performs error correction on the reads first
- Maps reads back to the contigs and scaffolds as a check
- Under active development

Switch to Hands-on :

Assembly with SPADES

Assessment of assembly quality

After assembly, we use QUAST (Gurevich, Saveliev, Vyahhi, and Tesler, 2013) to evaluate and compare genome assemblies.

What QUAST does :

- De novo genome assembly evaluation
- Reference-based evaluation
- Evaluating so-called misassemblies
- Report and visualisation

De novo metrics

Evaluation of the assembly based on

- Number of contigs greater than a given threshold (0, 500nct, 1kb)
- Total / threshold assembly size
- largest contig size
- N50 : the sequence length of the shortest contig at 50% of the total assembly length (equivalent to a median of contig lengths)
- L50 : the number of contigs at 50% of the total assembly length
- N75/L75 idem, for 75% of the assembly length

Reference-based metrics

- Metrics based on based on an alignment of all contigs on a reference genome. :
    - duplication rate
    - percent genome complete
    - NGA50 : equivalent of N50 but with the aligned block of the contigs
    - "Misassemblies" : breakpoint of alignment in a contigs. "
    - Visualisation available

Switch to Hands-on :

Assembly QC with Quast

Alignment

Alignment strategies

```
GAAGCTCTAGGATTACGATCTTGATCGCCGGGAAATTATGATCCTGACCTGAGTTTAAGGCATGGACCCATAA
                 ATCTTGATCGCCGAC----ATT       # GLOBAL
                 ATCTTGATCGCCGACATT           # LOCAL, with soft clipping
```

Global alignment

Global alignments, which attempt to align every residue in every sequence, are most useful when the sequences in the query set are similar and of roughly equal size. (This does not mean global alignments cannot start and/or end in gaps.) A general global alignment technique is the `Needleman–Wunsch algorithm`, which is based on dynamic programming.

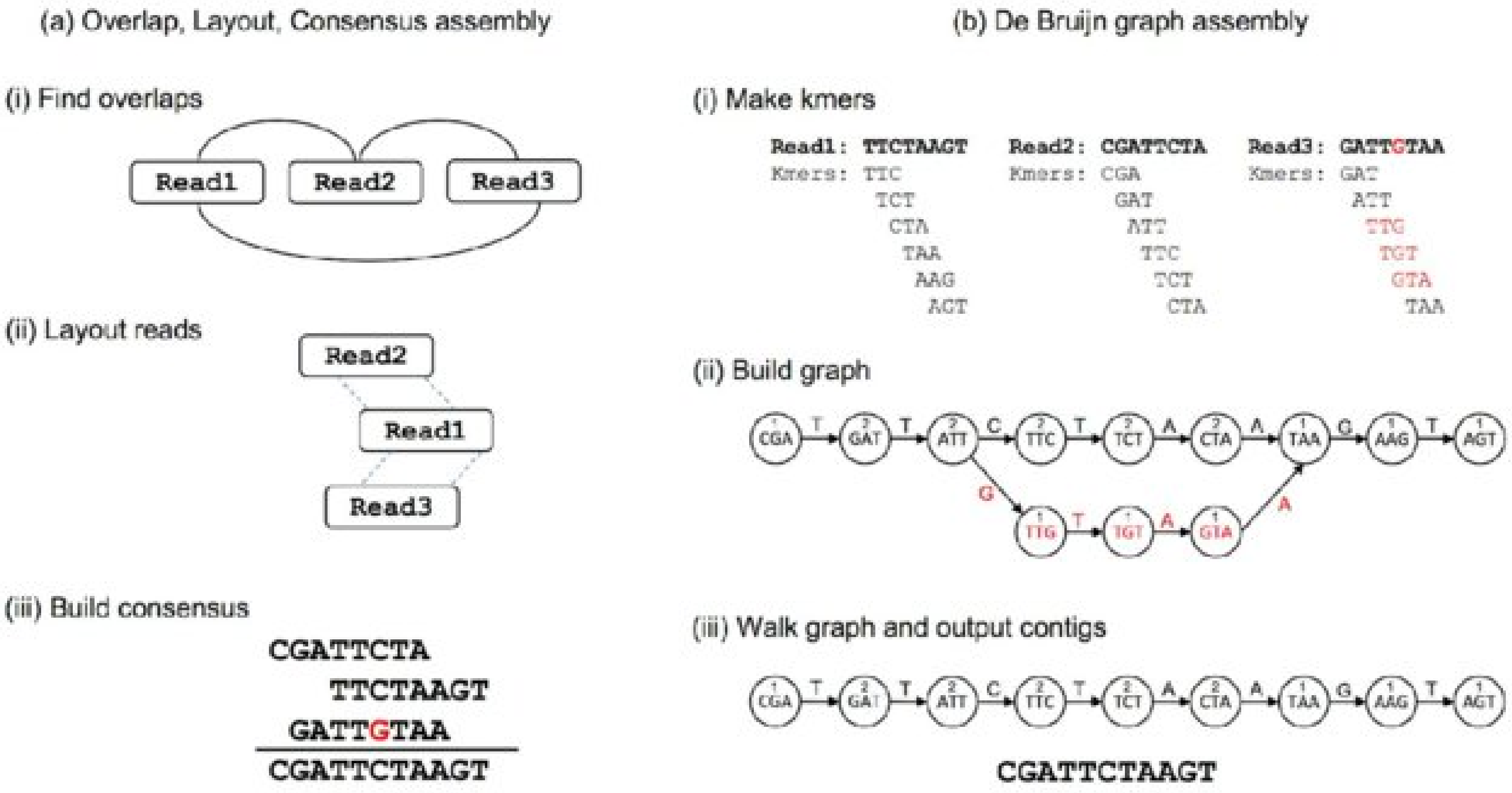
Alignment strategies

```
GAAGCTCTAGGATTACGATCTTGATCGCCGGGAAATTATGATCCTGACCTGAGTTTAAGGCATGGACCCATAA
                 ATCTTGATCGCCGAC----ATT       # GLOBAL
                 ATCTTGATCGCCGACATT           # LOCAL, with soft clipping
```

Local alignment

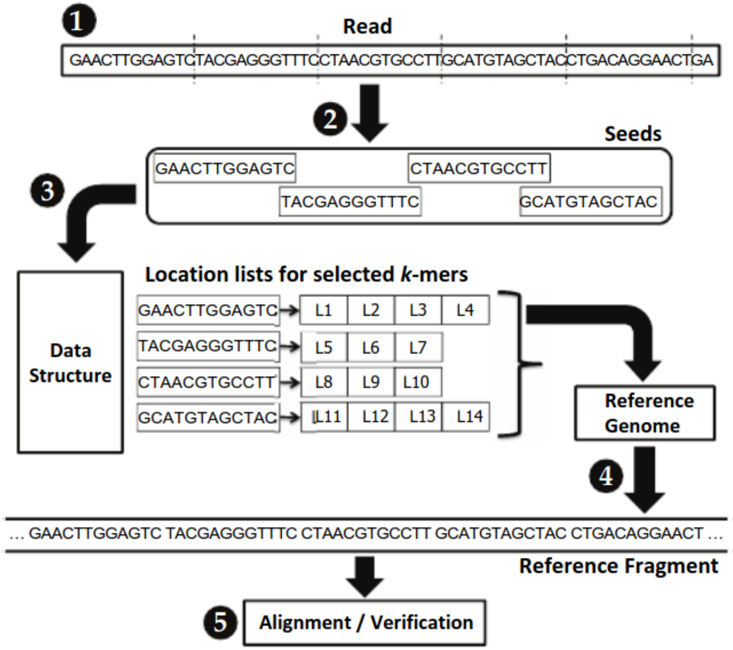
Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similarity or similar sequence motifs within their larger sequence context. The `Smith-Waterman algorithm` is a general local alignment method based on the same dynamic programming scheme but with additional choices to start and end at any place.

Seed-and-extend especially adapted to NGS data



Seed-and-extend especially adapted to NGS data

Seed-and-extend mappers are a class of read mappers that break down each read sequence into seeds (i.e., smaller segments) to find locations in the reference genome that closely match the read.


1
Read
GAACTTGGAGTCTACGAGGGTTTCCTAACGTGCCTTGCATGTAGCTACCTGACAGGAACTGA
2
Seeds
GAACTTGGAGTC
CTAACGTGCCTT
TACGAGGGTTTC
GCATGTAGCTAC
3
Data Structure
Location lists for selected k-mers
GAACTTGGAGTC → L1 L2 L3 L4
TACGAGGGTTTC → L5 L6 L7
CTAACGTGCCTT → L8 L9 L10
GCATGTAGCTAC → L11 L12 L13 L14
Reference Genome
4
… GAACTTGGAGTC TACGAGGGTTTC CTAACGTGCCTT GCATGTAGCTAC CTGACAGGAACT …
Reference Fragment
5
Alignment / Verification

1. The mapper obtains a read
2. The mapper selects smaller DNA segments from the read to serve as seeds
3. The mapper indexes a data structure with each seed to obtain a list of possible locations within the reference genome that could result in a match
4. For each possible location in the list, the mapper obtains the corresponding DNA sequence from the reference genome
5. The mapper aligns the read sequence to the reference sequence, using an expensive sequence alignment (i.e., verification) algorithm to determine the similarity between the read sequence and the reference sequence.

Mapping

- For further analysis it is necessary to map all the reads on the contigs.

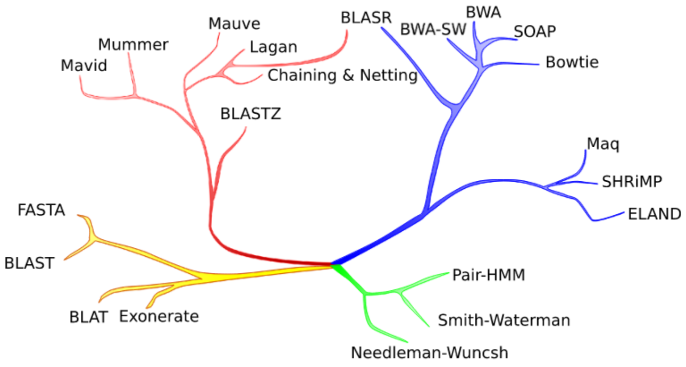


Figure 1 An illustration of relationships between alignment methods. The applications / corresponding computational restrictions shown are (green) short pairwise alignment / detailed edit model; (yellow) database search / divergent homology detection; (red) whole genome alignment / alignment of long sequences with structural rearrangements; and (blue) short read mapping / rapid alignment of massive numbers of short sequences. Although solely illustrative, methods with more similar data structures or algorithmic approaches are on closer branches. The BLASR method combines data structures from short read alignment with optimization methods from whole genome alignment.

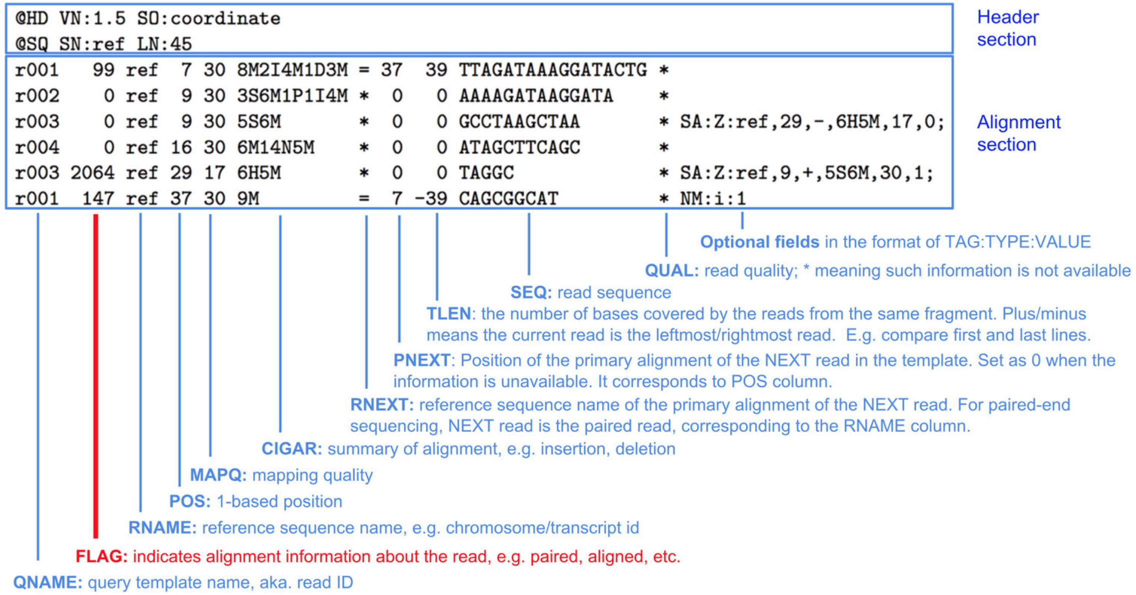
- We will use bowtie2 (Langmead and Salzberg, 2012)
  - Firstly, we build an index.
  - Secondly, reads are aligned.
  - We can use samtools (Li, Handsaker, Wysoker, Fennell, Ruan, Homer, Marth, Abecasis, and Durbin, 2009) and bedtools (Quinlan and Hall, 2010) to manipulate SAM/BAM files.

BAM/SAM

- SAM = Sequence Alignment Map
- BAM = Binary Alignment Map

These files represent an alignment of FASTQ reads against a reference like a FASTA.

- After a header section (for reference), each line represents the alignment of one read.



```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Optional fields in the format of TAG:TYPE:VALUE

QUAL: read quality; * meaning such information is not available

SEQ: read sequence

TLEN: the number of bases covered by the reads from the same fragment. Plus/minus means the current read is the leftmost/rightmost read. E.g. compare first and last lines.

PNEXT: Position of the primary alignment of the NEXT read in the template. Set as 0 when the information is unavailable. It corresponds to POS column.

RNEXT: reference sequence name of the primary alignment of the NEXT read. For paired-end sequencing, NEXT read is the paired read, corresponding to the RNAME column.

CIGAR: summary of alignment, e.g. insertion, deletion

MAPQ: mapping quality

POS: 1-based position

RNAME: reference sequence name, e.g. chromosome/transcript id

FLAG: indicates alignment information about the read, e.g. paired, aligned, etc.

QNAME: query template name, aka. read ID

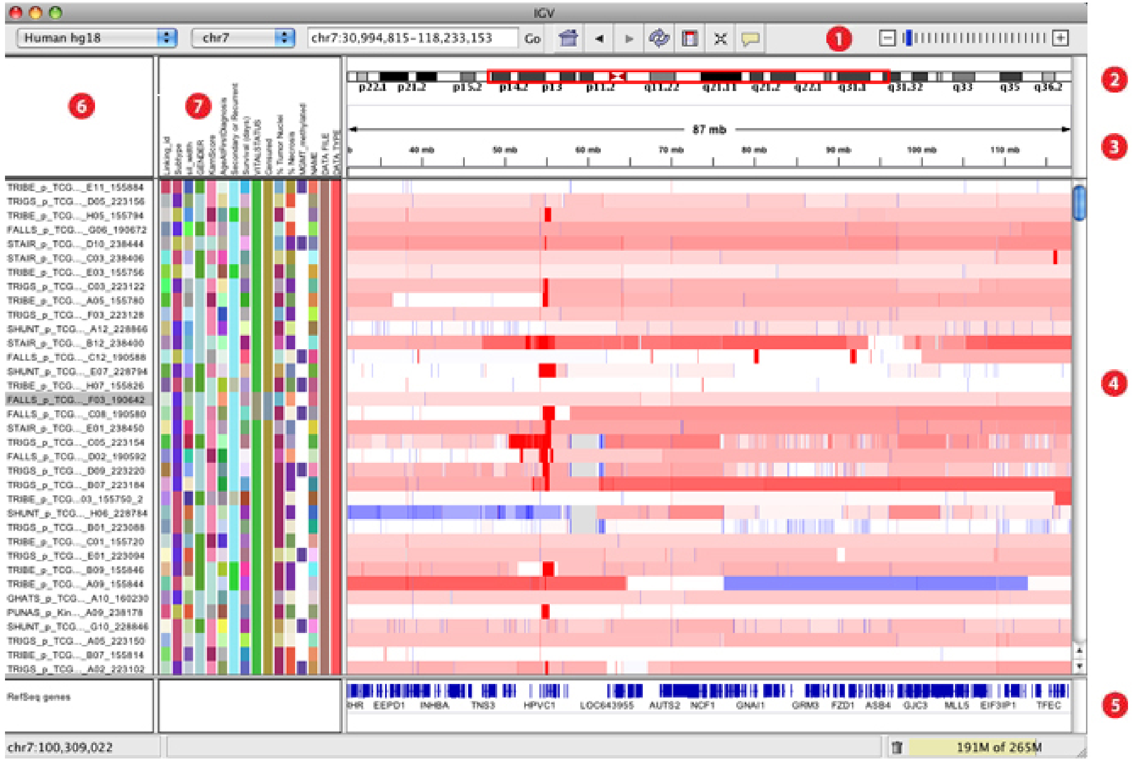
Switch to Hands-on :

Mapping

Visualization

Visualization

- Some tools for visualization and browsing
    - IGV (alignments and reference)
    - Artemis (genome and annotations)



Switch to Hands-on :

Visualization

Long reads

Tools for long reads

- Long read data can be used to improve assembly
- Bottlenecks :
    - DNA extraction (?)
    - cost of data generation
    - sequencing errors
- State of the art pipeline for assembly :
    - standalone long read assembly
    - FLYE (Kolmogorov, Rayko, Yuan, Polevikov, and Pevzner, 2019)
    - canu
    - Optional error correction with short reads
    - Unicycler

Take home message

Take home message

⟶ You have in your hands the first tools to analyze your NGS data

⟶ Data quality control is a crucial step

⟶ It is essential to define your plan analyses upstream of your project.

⟶ NGS are still an ongoing active bioinformatics research field

⟶ Biostatistics ...

References

Andrews, S. (2010). *FastQC A Quality Control tool for High Throughput Sequence Data*. URL: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

Bankevich, A., S. Nurk, D. Antipov, et al. (2012). "SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing". In: *Journal of computational biology* 19.5, pp. 455-477.

Goodwin, S., J. D. McPherson, and W. R. McCombie (2016). "Coming of age: ten years of next-generation sequencing technologies". In: *Nature Reviews Genetics* 17.6, pp. 333-351. DOI: 10.1038/nrg.2016.49. URL: https://doi.org/10.1038/nrg.2016.49.

Gurevich, A., V. Saveliev, N. Vyahhi, et al. (2013). "QUAST: quality assessment tool for genome assemblies". In: *Bioinformatics* 29.8, pp. 1072-1075.

Joshi, N. and J. Fass (2011). *Sickle: a sliding-window, adaptive, quality-based trimming tool for FastQ files*.

References(2)

Kolmogorov, M., M. Rayko, J. Yuan, et al. (2019). "metaFlye: scalable long-read metagenome assembly using repeat graphs".

DOI: 10.1101/637637. URL: https://doi.org/10.1101/637637.

Langmead, B. and S. L. Salzberg (2012). "Fast gapped-read alignment with Bowtie 2". In: *Nature Methods* 9.4, pp. 357-359. ISSN: 1548-7105. DOI: 10.1038/nmeth.1923. URL: http://dx.doi.org/10.1038/nmeth.1923.

Li, H., B. Handsaker, A. Wysoker, et al. (2009). "The sequence alignment/map format and SAMtools". In: *Bioinformatics* 25.16, pp. 2078-2079.

Martin, M. (2011). "Cutadapt removes adapter sequences from high-throughput sequencing reads". In: *EMBnet. journal* 17.1, pp. 10-12.

Quinlan, A. R. and I. M. Hall (2010). "BEDTools: a flexible suite of utilities for comparing genomic features". In: *Bioinformatics* 26.6, pp. 841-842.

References(3)

Zhou, Y., Y. Chen, S. Chen, et al. (2018). "fastp: an ultra-fast all-in-one FASTQ preprocessor". In: *Bioinformatics* 34.17, pp. i884-i890. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty560. eprint: http://academic.oup.com/bioinformatics/article-pdf/34/17/i884/25702346/bty560.pdf. URL: https://dx.doi.org/10.1093/bioinformatics/bty560.