First steps with NGS data

DUBii - Module 5

Valentin Loux - Olivier Rué

2020-03-09

Program

- Introduction (5 min)
- Get data from public resources (30 min)
- FASTQ format
- Quality control (45 min)
- Cleaning of reads (30 min)
- Mapping of reads (30 min)
- FASTA format
- SAM format

Preparation of your working directory

Instruction

- Go in your home directory
- Create a directory called M5 (i.e Module5) and move in
- Create this directory structure:

```
tree ~/M5
[orue@clust-slurm-client M5]$ tree .
.
├── CLEANING
├── FASTQ
├── MAPPING
└── QC

4 directories, 0 files
```

Correction

```
mkdir -p ~/M5/FASTQ
mkdir -p ~/M5/CLEANING
mkdir -p ~/M5/MAPPING
mkdir -p ~/M5/QC
cd ~/M5
```

The Data

What is data

Definition

- `Data` is *a symbolic representation of information*
- `Data` is stored in files whose format allows an easy way to access and manipulate
- `Data` represent the knowledge at a given time.

Properties

- The same information may be represented in different formats
- The content depends on technologies

> Understanding data formats, what information is encoded in each, and when it is appropriate to use one format over another is an essential skill of a bioinfor- matician.

Genomics sequences resources

The International Nucleotide Sequence Database Collaboration (INSDC) is a long-standing foundational initiative that operates between DDBJ, EMBL-EBI and NCBI. INSDC covers the spectrum of data raw reads, through alignments and assemblies to functional annotation, enriched with contextual information relating to samples and experimental configurations.

Data type	DDBJ	EMBL-EBI	NCBI
Next generation reads	Sequence Read Archive	European Nucleotide Archive (ENA)	Sequence Read Archive
Capillary reads	Trace Archive		Trace Archive
Annotated sequences	DDBJ		GenBank
Samples	BioSample		BioSample
Studies	BioProject		BioProject

INDSC resources

International Nucleotide Sequence Database Collaboration

The member organizations of this collaboration are:

- NCBI: National Center for Biotechnology Information
- EMBL: European Molecular Biology Laboratory
- DDBJ: DNA Data Bank of Japan

The INSDC has set up rules on the types of data that will be mirrored. The most important of these from a bioinformatician's perspective are:

- GenBank/Ebi ENA contains all annotated and identified DNA sequence information
- SRA NCBI Sequence Reads Archive / ENA European Nucleotide Archive: Short Read Archive contains measurements from high throughput sequencing experiments (raw data)

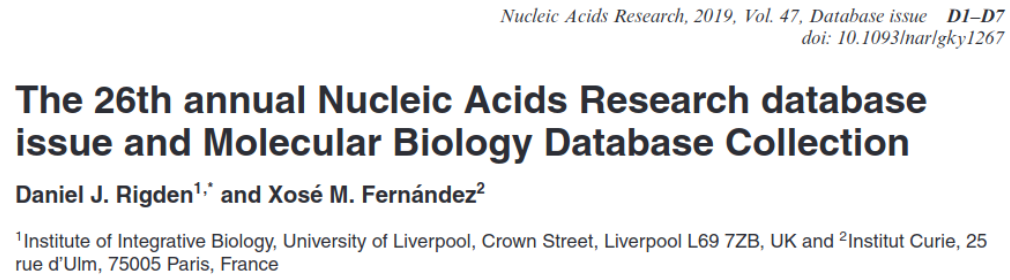
Deposit of sequencing (raw) and processed (analyzed) datas are (most of the time) a prerequiste for publication.

Other sequence resources

NAR Database Issue

Once a year the journal Nucleic Acids Research publishes its so-called “database issue”. Each article of this issue of the journal will provide an overview of generic and specific databases written by the maintainers of that resource.

- View the NAR: 2019 Database Issue.


Nucleic Acids Research, 2019, Vol. 47, Database issue ***D1–D7***
doi: 10.1093/nar/gky1267

The 26th annual Nucleic Acids Research database issue and Molecular Biology Database Collection

Daniel J. Rigden[1,*] and Xosé M. Fernández[2]

[1]Institute of Integrative Biology, University of Liverpool, Crown Street, Liverpool L69 7ZB, UK and [2]Institut Curie, 25 rue d'Ulm, 75005 Paris, France

NAR 2019 database issue overview

Getting raw data

Getting raw data

Sequencing data

- Specialized Tools or API are offered by the public repository to easily get data locally
- ENA: enaBrowserTools (command line, python, R)
- NCBI: sra-toolkit (command line, python, R)

Common command lines (wget) are most of the time also available

Hands-on: Getting raw data

Instruction

Get the raw shot read data (Illumina) associated with this article (Allué-Guardia, Nyong, Koenig, Vargas, Bono, and Eppinger, 2019).

Genome Sequences

Closed Genome Sequence of *Escherichia coli* K-12 Group Strain C600

Anna Allué-Guardia, Emmanuel C. Nyong, Sara S. K. Koenig, Sean M. Vargas, James L. Bono, Mark Eppinger

Julia A. Maresca, Editor

DOI: 10.1128/MRA.01052-18 Check for updates

- In the "Data availability" section, extract the accession for Illumina data : SRX4909245
- Explore SRA and ENA

Get the data by the method of your choice:

- use `wget` or `fasterq-dump` from `sra-tools`

Compress FASTQ files with `gzip`

Correction

- Direct download via the web brower
- Using wget :

```
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR808/003/SRR8082143/SRR8082143_1.fastq.gz
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR808/003/SRR8082143/SRR8082143_2.fastq.gz
```

- Using sra-toolkit

```
module load  sra-tools
srun fasterq-dump -S -p SRR8082143 --outdir . --threads 1
```

- enaBrowserTool is also available
- Compress FASTQ files

```
gzip *.fastq
```

```
ls -ltrh ~/M5/FASTQ/
total 236M
-rw-rw-r-- 1 orue orue 127M  6 mars  12:32 SRR8082143_2.fastq.gz
-rw-rw-r-- 1 orue orue 109M  6 mars  12:32 SRR8082143_1.fastq.gz
```

Sequencing - Vocabulary

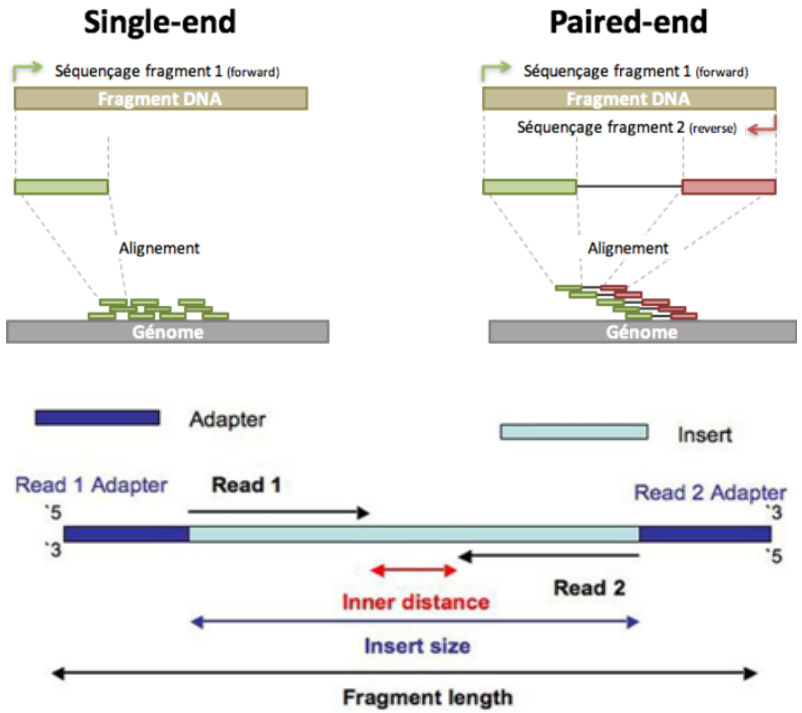
Read : piece of sequenced DNA

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

Insert = Fragment size

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

Coverage = % of genome covered



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**.

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.

Example

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

FASTQ quality

The weird characters in the 4th section are the so called “encoded” numerical values. In a nutshell, 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
```

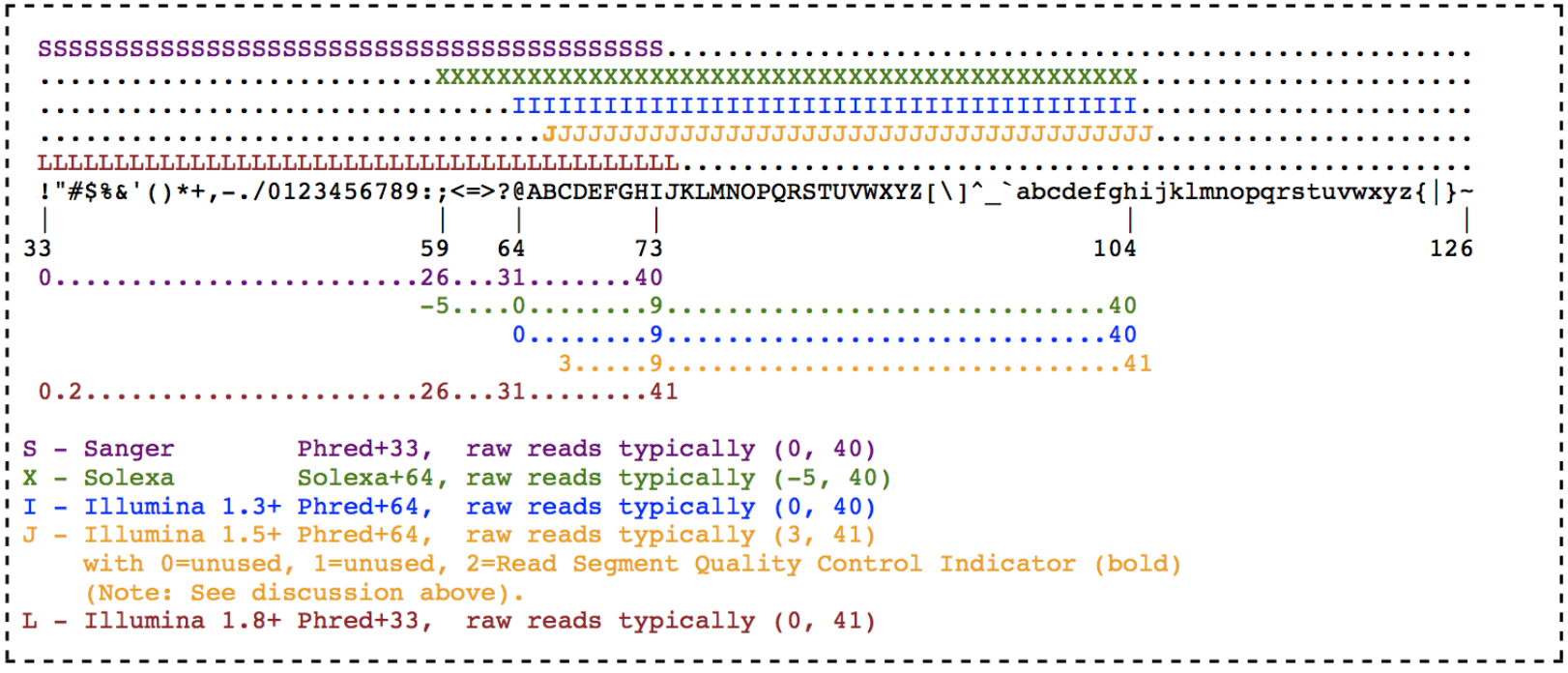
The quality values of the FASTQ files are on top. The numbers in the middle of the scale from 0 to 40 are called Phred scores. The numbers represent the error probabilities via the formula:

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.



FASTQ encoding values

FASTQ toolbox

seqtk

Seqtk (Li, 2012) is a fast and lightweight tool for processing sequences in the FASTA or FASTQ format. It seamlessly parses both FASTA and FASTQ files which can also be optionally compressed by gzip.

```
module load seqtk
seqtk

Usage:   seqtk <command> <arguments>
Version: 1.3-r106

Command: seq       common transformation of FASTA/Q
         comp      get the nucleotide composition of FASTA/Q
         sample    subsample sequences
         subseq    extract subsequences from FASTA/Q
         fqchk     fastq QC (base/quality summary)
         mergepe   interleave two PE FASTA/Q files
         trimfq    trim FASTQ using the Phred algorithm
         hety      regional heterozygosity
         gc        identify high- or low-GC regions
         mutfa     point mutate FASTA at specified positions
         mergefa   merge two FASTA/Q files
         famask    apply a X-coded FASTA to a source FASTA
         dropse    drop unpaired from interleaved PE FASTA/Q
         rename    rename sequence names
         randbase  choose a random base from hets
         cutN      cut sequence at long N
         listhet   extract the position of each het
```

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.

Quality control

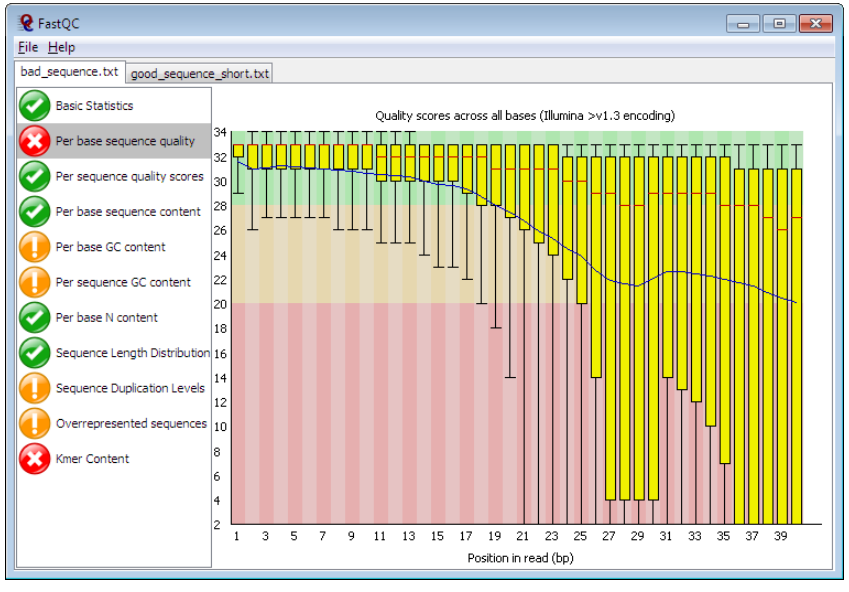
Why QC'ing your reads ?

Try to answer to (not always) simple questions :

- Do the generated sequences conform to the expected level of performance?
  - Size
  - Number of reads
  - Quality
- Residual presence of adapters or indexes ?
- Are there (un)expected techincal biases
- Arte ther (un)expected biological biases

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



FastQC software

Hands-on : Quality control

Instruction

- Launch FastQC on the paired-end FastQ files of the sample you previously downloaded
- Inspect the results
  - Are the number coherent with the article ?
  - Comment on the quality of the sequencing

Correction

```
cd ~
module load fastqc
srun --cpus-per-task 8 fastqc FASTQ/SRR8082143_1.fastq.gz -o QC/ -t 8
srun --cpus-per-task 8 fastqc FASTQ/SRR8082143_2.fastq.gz -o QC/ -t 8
```

```
ls -ltrh ~/M5/QC
total 1,9M
-rw-rw-r-- 1 orue orue 321K  6 mars  13:23 SRR8082143_1_fastqc.zip
-rw-rw-r-- 1 orue orue 642K  6 mars  13:23 SRR8082143_1_fastqc.html
-rw-rw-r-- 1 orue orue 333K  6 mars  13:23 SRR8082143_2_fastqc.zip
-rw-rw-r-- 1 orue orue 642K  6 mars  13:23 SRR8082143_2_fastqc.html
```

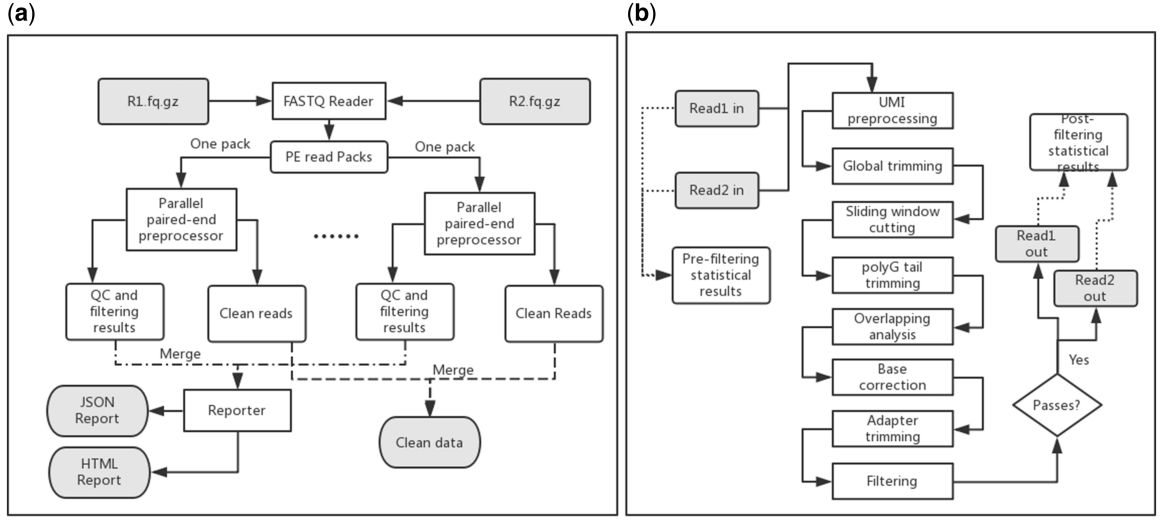
Reads cleaning

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

Hands-on : reads cleaning with fastp

Instruction

- Launch fastp on the paired-end FastQ files of the sample you previously downloaded
    - Detect and Remove the classical Illumina adapters
    - Filter reads with :
        - mean quality >= 20 on a sliding window of 4
        - 40% of the bases with a quality >= 15
        - length of the trimmed read >= 100
- Inspect the results
    - How many reads are filtered ?
    - Where do fastp store its reports. Is it configurable ?

Correction

```
module load fastp
cd ~/M5
srun --cpus-per-task 8 fastp \
 --in1 FASTQ/SRR8082143_1.fastq.gz \
 --in2 FASTQ/SRR8082143_2.fastq.gz \
 -l 100 \
 --out1 CLEANING/SRR8082143_1.cleaned_filtered.fastq.gz \
 --out2 CLEANING/SRR8082143_2.cleaned_filtered.fastq.gz \
 --unpaired1 CLEANING/SRR8082143_singles.fastq.gz \
 --unpaired2 CLEANING/SRR8082143_singles.fastq.gz \
 -w 1 \
 -j CLEANING/fastp.json \
 -h CLEANING/fastp.html \
 -t 8
```

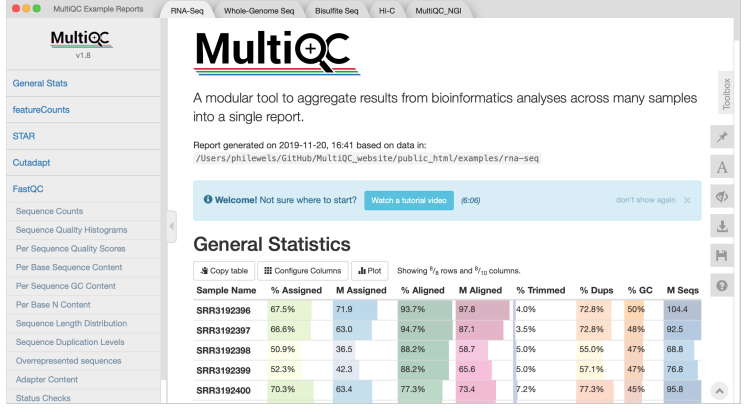
```
ls -ltrh ~/M5/CLEANING/
total 245M
-rw-rw-r-- 1 orue orue 113M  6 mars  12:59 SRR8082143_1.cleaned_filtered.fastq.gz
-rw-rw-r-- 1 orue orue 162K  6 mars  12:59 fastp.json
-rw-rw-r-- 1 orue orue 525K  6 mars  12:59 fastp.html
-rw-rw-r-- 1 orue orue 2,2M  6 mars  12:59 SRR8082143_singles.fastq.gz
-rw-rw-r-- 1 orue orue 130M  6 mars  12:59 SRR8082143_2.cleaned_filtered.fastq.gz
```

One report to rule them all

MultiqQC (Ewels, Magnusson, Lundin, and Käller, 2016) allow the aggregation of individual reports from FastQC, Fastp, Trimmomactic, Cutadapt and much more

- 78 tools included
- Aggregate all analysis in one report :
  - by tool
  - in one graphe aggregating samples

MultiQC tools



MultiQC Report Example

Hands-on: MultiQC

Instruction

Run MultiQC to obtain a report with fastqc and fastp results

Correction

```
cd ~/M5
module load multiqc
multiqc -d . -o CLEANING
```

```
ls -ltrh ~/M5/CLEANING/
total 248M
-rw-rw-r-- 1 orue orue 113M  6 mars  12:59 SRR8082143_1.cleaned_filtered.fastq.gz
-rw-rw-r-- 1 orue orue 162K  6 mars  12:59 fastp.json
-rw-rw-r-- 1 orue orue 525K  6 mars  12:59 fastp.html
-rw-rw-r-- 1 orue orue 2,2M  6 mars  12:59 SRR8082143_singles.fastq.gz
-rw-rw-r-- 1 orue orue 130M  6 mars  12:59 SRR8082143_2.cleaned_filtered.fastq.gz
-rw-rw-r-- 1 orue orue 1,2M  6 mars  13:28 multiqc_report.html
drwxrwxr-x 2 orue orue 2,0M  6 mars  13:28 multiqc_data
```

Mapping

Mapping

- Map short reads to a reference genome is predict the locus where a read comes from.
- The result of a mapping is the list of the most probable regions with an associated probability.

But what is a reference?

Reference

It can be everything containing DNA information:

- Complete genome
- Assembly
- Set of contigs
- Set of sequences
- Genes, non-coding RNA…

For mapping, references have to be stored in a `FASTA` file.

FASTA format

Informations inside

The FASTA format is used to represent sequence information. The format is very simple:

- A `>` symbol on the FASTA header line indicates a fasta record start.
- A string of letters called the sequence id may follow the `>` symbol.
- The header line may contain an arbitrary amount of text (including spaces) on the same line.
- Subsequent lines contain the sequence.

Example

```
>foo
ATGCC
>bar other optional text could go here
CCGTA
>bidou
ACTGCAGT
TTCGN
>repeatmasker
ATGTGTcgggggATTTT
>prot2; my_favourite_prot
MTSRRSVKSGPREVPRDEYEDLYYTPSSGMASP
```

FASTA syntax

The lack of a definition of the FASTA format and its apparent simplicity can be a source of some of the most confounding errors in bioinformatics. Since the format appears so exceed- ingly straightforward, software developers have been tacitly assuming that the properties they are accustomed to are required by some standard - whereas no such thing exists.

Common problems

- Some tools need 60 characters per line
- Some tools ignore anything following the first space in the header line
- Some tools are very restrictive on the alphabet used
- Some tools require uppercase letters

FASTA formating

Good practices

The sequence lines should always wrap at the same width (with the exception of the last line). Some tools will fail to operate correctly and may not even warn the users if this condition is not satisfied. The following is technically a valid FASTA but it may cause various subtle problems.

```
>foo
ATGCATGCATGCATGCATGC
ATGCATGCA
TGATGCATGCATGCATGCA
```

should be reformated to

```
>foo
ATGCATGCATGCATGCATGC
ATGCATGCATGATGCATGCA
TGCATGCA
```

Can be easily to with seqkit (Shen, Le, Li, and Hu, 2016)

```
seqkit seq -w 60 seqs.fa > seqs2.fa
```

FASTA Header

Some data repositories will format FASTA headers to include structured information. Tools may operate differently when this information is present in the FASTA header. Below is a list of the recognized FASTA header formats.

Type	Format(s) [1]	Example(s)
local	lcl|integer lcl|string	lcl|123 lcl|hmm271
GenInfo backbone seqid	bbs|integer	bbs|123
GenInfo backbone moltype	bbm|integer	bbm|123
GenInfo import ID	gim|integer	gim|123
GenBank	gb|accession|locus	gb|M73307|AGMA13GT
EMBL	emb|accession|locus	emb|CAM43271.1|
PIR	pir|accession|name	pir||G36364
SWISS-PROT	sp|accession|name	sp|P01013|OVAX_CHICK
patent	pat|country|patent|sequence	pat|US|RE33188|1
pre-grant patent	pgp|country|application-number|seq-number	pgp|EP|0238993|7
RefSeq [2]	ref|accession|name	ref|NM_010450.1|
general database reference	gnl|database|integer gnl|database|string	gnl|taxon|9606 gnl|PID|e1632
GenInfo integrated database	gi|integer	gi|21434723
DDBJ	dbj|accession|locus	dbj|BAC85684.1|
PRF	prf|accession|name	prf||0806162C
PDB	pdb|entry|chain	pdb|1I4L|D
third-party GenBank	tpg|accession|name	tpg|BK003456|
third-party EMBL	tpe|accession|name	tpe|BN000123|
third-party DDBJ	tpd|accession|name	tpd|FAA00017|

FASTA header examples

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.

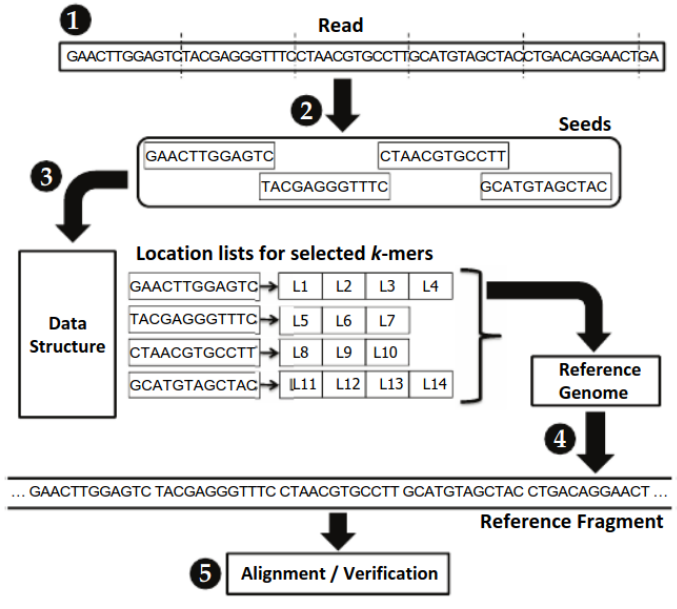
Local alignment

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 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. First, the mapper obtains a read
2. Second, the mapper selects smaller DNA segments from the read to serve as seeds
3. Third, 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. Fourth, for each possible location in the list, the mapper obtains the corresponding DNA sequence from the reference genome
5. Fifth, 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 tools

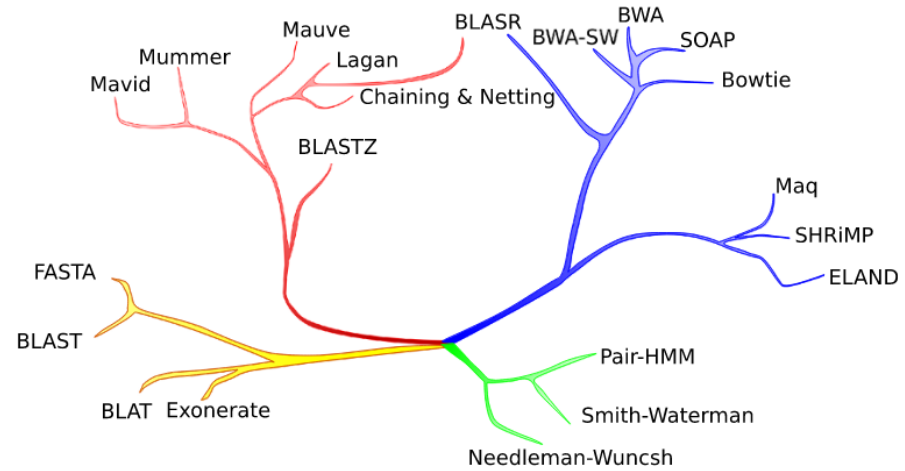


Figure 1 An illustration of relationships between alignment methods. The applications / corresponding computational restrictions shown are (green) short pairwise alignment / detailed edit model; (yellow) database search / divergent homology detection; (red) whole genome alignment / alignment of long 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.

- Short reads: BWA (Li, 2013)/ BOWTIE (Langmead and Salzberg, 2012)

Hands-on: mapping with bwa

Instruction

- Map the reads to the reference genome `/shared/projects/dubii2020/data/module5/seance1/CP031214.1.fasta` with `bwa`

Correction

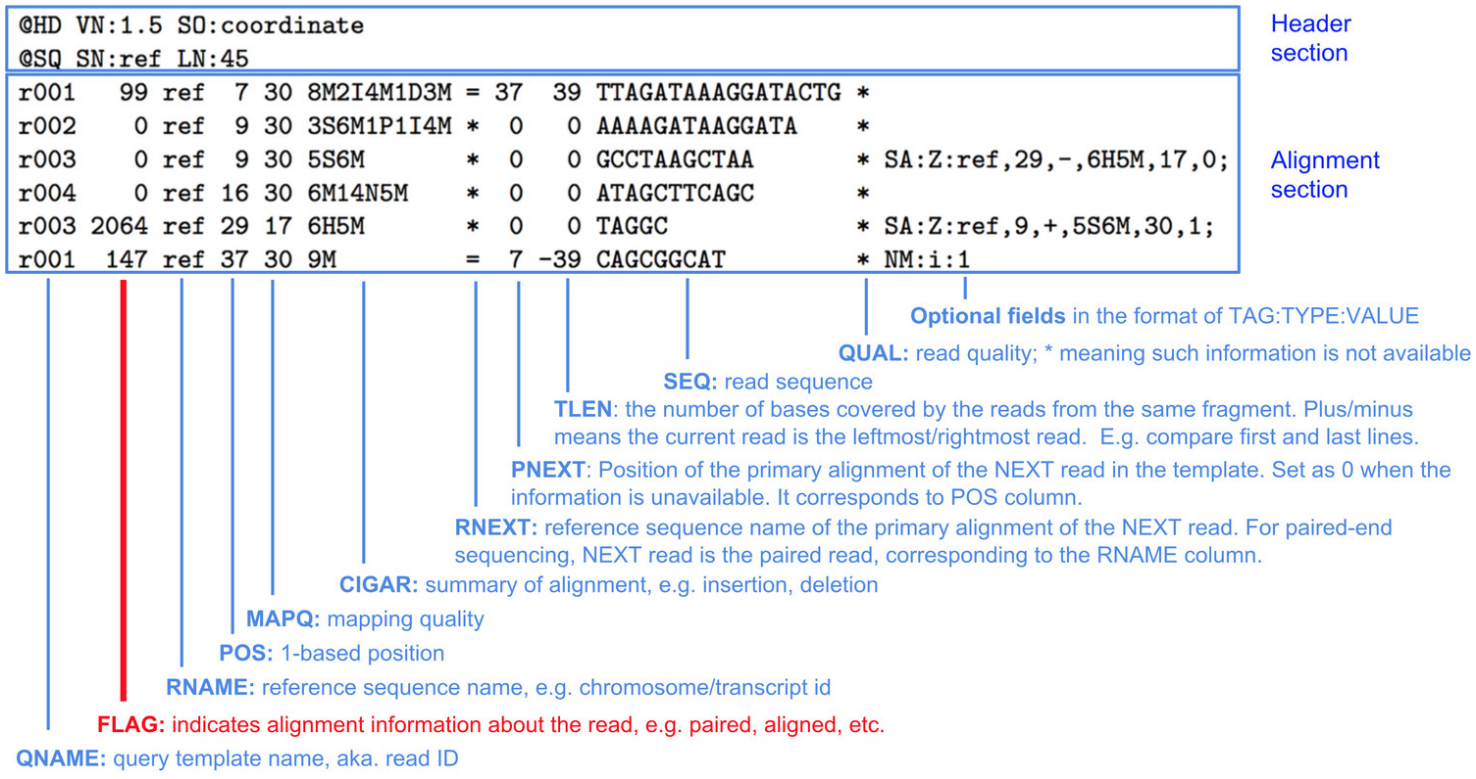
```
cd ~/M5
module load bwa
# srun bwa index sequence.fasta
srun --cpus-per-task=33 bwa mem \
  /shared/projects/dubii2020/data/module5/seance1/CP031214.1.fasta \
  CLEANING/SRR8082143_1.cleaned_filtered.fastq.gz \
  CLEANING/SRR8082143_2.cleaned_filtered.fastq.gz \
  -t 32 \
  | \
  samtools view -hbS - > MAPPING/SRR8082143.bam
```

```
ls -ltrh ~/M5/MAPPING/
total 249M
-rw-rw-r-- 1 orue orue 249M  6 mars  13:01 SRR8082143.bam
```

Sequence Alignment Format (SAM)

SAM / BAM formats

The SAM/BAM formats are so-called Sequence Alignment Maps. These files typically represent the results of aligning a FASTQ file to a reference FASTA file and describe the individual, pairwise alignments that were found. Different algorithms may create different alignments (and hence BAM files)



SAM FLAG

FLAGS contain a lot of informations.

Binary (Decimal)	Hex	Description
00000000001 (1)	0x1	Is the read paired?
00000000010 (2)	0x2	Are both reads in a pair mapped "properly" (i.e., in the correct orientation with respect to one another)?
00000000100 (4)	0x4	Is the read itself unmapped?
00000001000 (8)	0x8	Is the mate read unmapped?
00000010000 (16)	0x10	Has the read been mapped to the reverse strand?
00000100000 (32)	0x20	Has the mate read been mapped to the reverse strand?
00001000000 (64)	0x40	Is the read the first read in a pair?
00010000000 (128)	0x80	Is the read the second read in a pair?
00100000000 (256)	0x100	Is the alignment not primary? (A read with split matches may have multiple primary alignment records.)
01000000000 (512)	0x200	Does the read fail platform/vendor quality checks?
10000000000 (1024)	0x400	Is the read a PCR or optical duplicate?

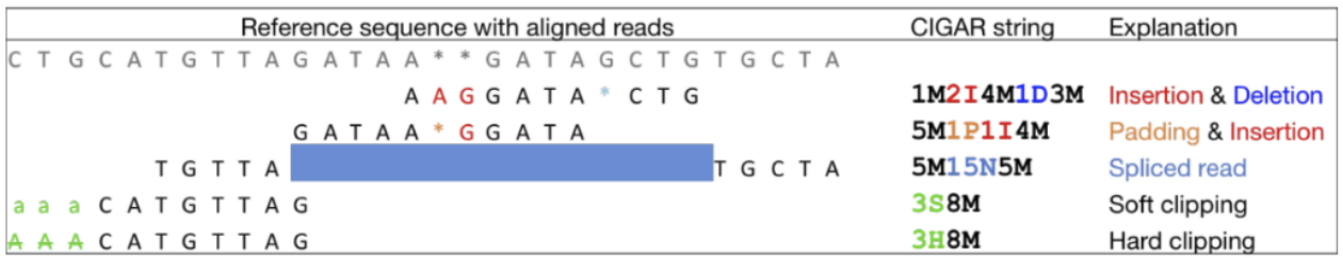
SAM CIGAR

`CIGAR` stands for *Concise Idiosyncratic Gapped Alignment Report*. This sixth field of a SAM file contains a so-called CIGAR string indicating which operations were necessary to map the read to the reference sequence at that particular locus.

The following operations are defined in CIGAR format (also see figure below):

- **M** - Alignment (can be a sequence match or mismatch!)
- **I** - Insertion in the read compared to the reference
- **D** - Deletion in the read compared to the reference
- **N** - Skipped region from the reference. For mRNA-to-genome alignments, an N operation represents an intron. For other types of alignments, the interpretation of N is not defined.
- **S** - Soft clipping (clipped sequences are present in read); S may only have H operations between them and the ends of the string
- **H** - Hard clipping (clipped sequences are NOT present in the alignment record); can only be present as the first and/or last operation
- **P** - Padding (silent deletion from padded reference)
- **=** - Sequence match (not widely used)
- **X** - Sequence mismatch (not widely used)

The sum of lengths of the **M**, **I**, **S**, **=**, **X** operations must equal the length of the read. Here are some examples:



SAM toolbox

Samtools & Picard tools

Samtools (Li, Handsaker, Wysoker, Fennell, Ruan, Homer, Marth, Abecasis, and Durbin, 2009) and Picard tools (Broad Institute, 2018) are Swiss-knifes for operating of SAM/BAM format

- Visualize
- Filter
- Stats
- Index
- Merge
- ...

Some examples with samtools

```
# Visualize BAM content in SAM format
samtools view -h MAPPING/SRR8082143.bam
# Sort BAM file
samtools sort MAPPING/SRR8082143.bam -o MAPPING/SRR8082143.sorted.bam
# Index sorted BAM file
samtools index MAPPING/SRR8082143.sorted.bam
# Get some statistics
samtools flagstat MAPPING/SRR8082143.sorted.bam
# Extract specific region
samtools view MAPPING/SRR8082143.sorted.bam CP031214.1:1-1000
# Extract specific region in a BAM file
samtools view -h MAPPING/SRR8082143.sorted.bam CP031214.1:1-1000 |samtools view -bS -  > MAPPING/SRR8082143.1-1000.bam
# ...
```

Picard tools

```
module load picard
picard -h
```

What about Long Reads ?

As global quality and error models ar different, ,algorithms and tools are different for long reads.

The raw read format is also different

- PacBio :
  - internal read correction
  - built in software for QC / correction
  - QC : nanoPlot (De Coster, D'Hert, Schultz, Cruts, and Van Broeckhoven, 2018)
  - Correction (hybrid) : LorDec (Salmela and Rivals, 2014)
  - Alignment minimap2 (Li, 2018), BLASR (Chaisson and Tesler, 2012)
- NanoPore :
  - Caution to basecaller / chemistry version !
  - QC : nanoPlot
  - Correction : Canu (Koren, Walenz, Berlin, Miller, Bergman, and Phillippy, 2017), MECAT (Xiao, Chen, Xie, Chen, Wang, Han, Luo, and Xie, 2017)

References

Allué-Guardia, A, E. C. Nyong, S. S. K. Koenig, et al. (2019). "Closed Genome Sequence of Escherichia coli K-12 Group Strain C600". In: *Microbiology Resource Announcements* 8.2. Ed. by J. A. Maresca. DOI: 10.1128/MRA.01052-18. eprint: https://mra.asm.org/content/8/2/e01052-18.full.pdf. URL: https://mra.asm.org/content/8/2/e01052-18.

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

Broad Institute (2018). *Picard Tools*. .

Chaisson, M. J. and G. Tesler (2012). "Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory". In: *BMC bioinformatics* 13.1, p. 238.

De Coster, W, S. D'Hert, D. T. Schultz, et al. (2018). "NanoPack: visualizing and processing long-read sequencing data". In: *Bioinformatics* 34.15, pp. 2666-2669. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty149. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/15/2666/25230836/bty149.pdf. URL: https://doi.org/10.1093/bioinformatics/bty149.

Ewels, P, M. Magnusson, S. Lundin, et al. (2016). "MultiQC: summarize analysis results for multiple tools and samples in a single report". In: *Bioinformatics* 32.19, pp. 3047-3048.

References

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

Koren, S, B. P. Walenz, K. Berlin, et al. (2017). "Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation". In: *Genome research* 27.5, pp. 722-736.

Langmead, B. and S. Salzberg (2012). *Fast gapped-read alignment with bowtie 2 Nat Methods 9 (4): 357-359. pmid: 22388286 View Article PubMed*.

Li, H. (2012). *seqtk Toolkit for processing sequences in FASTA/Q formats*.

Li, H. (2013). "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM". In: *arXiv preprint arXiv:1303.3997*.

Li, H. (2018). "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094-3100.

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.

References

Salmela, L. and E. Rivals (2014). "LoRDEC: accurate and efficient long read error correction". In: *Bioinformatics* 30.24, pp. 3506-3514.

Shen, W, S. Le, Y. Li, et al. (2016). "SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation". In: *PloS one* 11.10.

Xiao, C, Y. Chen, S. Xie, et al. (2017). "MECAT: fast mapping, error correction, and de novo assembly for single-molecule sequencing reads". In: *nature methods* 14.11, p. 1072.

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.