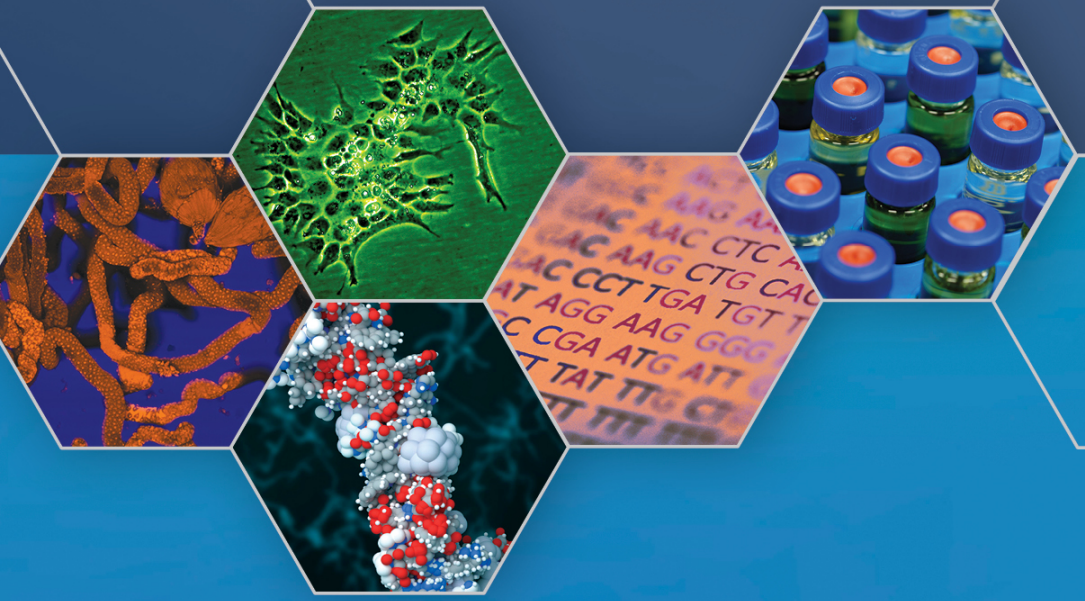
ICBR

Interdisciplinary Center for Biotechnology Research



UF | UNIVERSITY of FLORIDA

ICBR

Interdisciplinary Center for Biotechnology Research

Bioinformatics 101 – Lecture 4

Basic Next-Gen Analysis

Alberto Riva (ariva@ufl.edu), J. Lucas Boatwright (jlboat@ufl.edu)

ICBR Bioinformatics Core

UF | UNIVERSITY of FLORIDA

Short Reads - Structure

Short reads are saved in **fastq** files. Each read is represented by four lines of text:

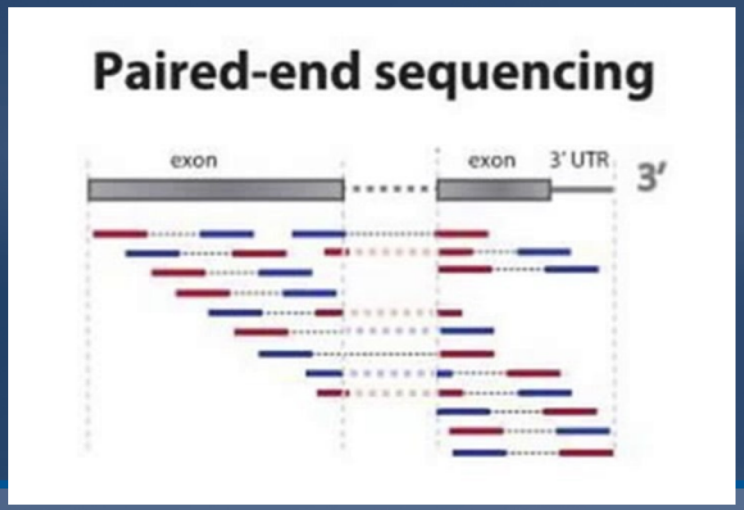
```
@M01105:32:000000000-BFW77:1:1101:15747:1360 1:N:0:CATGGC
CTTACTAATACATCCCCTTCCTTAAACCCACCAAAACTTTTCCAACTTTCCTCTCTT
+
6BC9BF<,C,C<FE@EDFGEFFF9,,AAC,@>8,,,;@,,,;;,,0;,,34;=<?EE
```

1. Read header
2. Sequence
3. (empty)
4. Quality scores

Short Reads - Structure

- Read length can range from 35 to 300nt; a typical fastq file may contain tens of millions of reads, and be several GB in size.

- Fastq files may contain reads from different samples (**multiplexing**); **index sequences** are used to separate them.



- Paired-end sequencing produces two fastq files, one containing R1 reads, the other containing the corresponding R2 reads.

Short Reads - Structure

Paired reads:

```
@M01105:32:000000000-BFW77:1:1101:15747:1360 1:N:0:CATGGC
CTTACTAATACATCCCCTTCCTTAAACCCACCAAAACTTTTCCAACTTTCCTCTCTT
+
6BC9BF<,C,C<FE@EDFGEFFF9,,AAC,@>8,,,;@,,,;;,,0;,,34;=<?EE
```

```
@M01105:32:000000000-BFW77:1:1101:15747:1360 2:N:0:CATGGC
TTCATCCTCCTCTATTTCTTACTCTTCCTCTTGCCTTTTCTGCCCCGTCCCCTCGTC
+
6,8,,<,,;,,6;,,,,,;,,,,;,4;44=,3,,,,,,,,+,33****)*1)*0)58
```

Short Reads – Quality Scores

- Each character represents a quality value.
- Sanger encoding is now the most commonly used.

```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 ..................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
 !"#$%&'()*+,-./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)
```

Short Reads – Quality Scores

Interpretation:

$$Q = -10 \log(p)$$

where p is the probability that the base is incorrect.

- Highest possible value is 40 (or 41), p = 0.0001.
- Values above 30 are acceptable.
- Bases with Q < 20 should never be used.
- Reads are **trimmed** to ensure average / minimum quality.

Short Reads – Quality Scores

Understanding the **error profile** of your reads is very important. Different technologies have different error profiles. For example:

- **Illumina** – Quality quickly reaches max, then slowly decreases towards 3' end of read.
- **PacBio** – Higher error rate but with uniform distribution across read.

Short Reads – Quality Scores

Why are quality scores important?

Certainty about a base call can only be obtained by observing that base multiple times (**coverage**).

How many times? Depends on the application.

PacBio sequencing has built-in repeated observations through CCS (Circular Consensus Sequencing) – allows trading read length for higher quality. E.g. a 20kb read can become a 5kb read with 4X coverage.

Short Reads – Quality Scores

Examples of tools to work with quality scores:

- trimmomatic
- cutadapt
- trim_galore
- sickle
- FastQC

These tools are usually executed automatically at the beginning of each analysis. *Pipelines* help make data analysis automated -> faster, reliable, reproducible.

Pipelines

- A *pipeline* is a tool to perform a complete analysis (end-to-end).
- It normally consists of multiple steps to be performed in sequence. Each step may be local or (more often) require submitting parallel jobs.
- A pipeline manager should handle job submission, processing of the results, generation of final report.

Simple pipelines

- Simple pipelines can be created by concatenating jobs to perform consecutive processing steps.
- For example: after previous script (script1.qsub), add step to count number of lines in output (script2.qsub).

```
#!/bin/bash

#SBATCH --time=10:00
#SBATCH --ntasks=3
#SBATCH --mem=1G

FILE=$1
OUT=$2
wc -l $FILE > $OUT
```

Simple pipelines

- We can now submit these two scripts as two separate jobs, scheduling one after the other.

```
$ submit script1.qsub genes.csv chroms.txt
1234567

$ submit -after 1234567 script2.qsub chroms.txt
```

- The second job will be "held" until the first one has terminated successfully, then it will be scheduled for running.

Simple pipelines

This approach only works in very simple cases.
Limitations:

- We had to come up with a name for the intermediate file – the pipeline manager should do that.
- What if the second step needs input from more than one previous step?
- If the first step fails to produce its output, the pipeline should stop.

Pipeline managers

Modern pipeline frameworks provide all the features mentioned above, and more. Example: **nextflow**. Features:

- Reproducible – workflows can run Docker/Singularity containers -> easy management of software versions.
- Portable – designed to interface with job schedulers or run locally;
- Parallelization – inherently parallel with input and output defining serial or parallel runs;
- Continuous checkpoints – can resume execution at last step after stopping.

```
process Fastqc {
    publishDir "results"

    input:
    set dataset_id, file(reads) from fastqc_paired_fastq

    output:
    file "*_fastqc.{zip,html}" into fastqc_results

    clusterOptions = { "--account=bioinf_workshop
                        --qos=bioinf_workshop --time=4:00:00
                        --mem-per-cpu=2gb --cpus-per-task=1" }
    module 'fastqc'

    script:
    """
    fastqc -q $reads
    """
}
```

UF | ICBR BIOINFORMATICS

UFHealth CANCER CENTER