INTRODUCTION TO BIOINFORMATICS

PRESENTED BY: PRAISSY ZEFI J (DTU)
CONTACT: pzeje@dtu.dk

LEARNING OBJECTIVES

- Understanding sequencing output and files generated
- File storing, naming, sharing and protection guidelines
- Introduction to Operating systems
- Introduction to Bioinformatics
- Evading conflict issues using docker containers, virtual machines and environments

SEQUENCING OUTPUT

Data generated after sequencing DNA → in the forms of sequence with quality scores

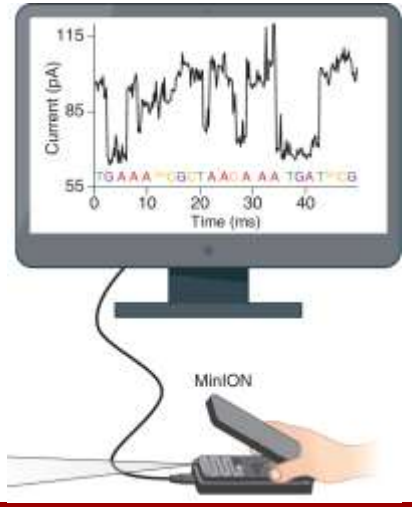
Can be in the form of chromatograms, text files or tables → files will be raw reads, mapped reads, aligned reads and corresponding quality scores

Analyzed using bioinformatic tools to interpret it biologically

ILLUMINA → FASTQ format (file123.fq)

ONT → FAST5 , POD5 format, FASTQ format

```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
@ERR000589.42 EAS139_45:5:1:2:1293/1
AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAAT
+
IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```



ILLUMINA SEQUENCING – OUTPUT FILE

FASTQ file: text-based sequencing data file format → stores raw sequence data and quality scores viewed using text-based editors or command line systems

SEQUENCE IDENTIFIER

SEQUENCE

SEPARATOR

QUALITY SCORES

```
1  @ERR000589.41 EAS139_45:5:1:2:111/1
2  CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
3  +
4  3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
5  @ERR000589.42 EAS139_45:5:1:2:1293/1
6  AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAAT
7  +
8  IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

FASTQ ORA Sequence File Format →
binary compressed file format of the text-based FASTQ file format
Up to 5x smaller than corresponding fastq.gz files without compromising data integrity

Quality score, Q = -10log10(e)

e → estimated probability that base call is wrong

(Q20) → error rate of 1 in 100 (meaning every 100 bp sequencing read may contain an error) i.e. accuracy = 99%.

QUALITY SCORE	ERROR PROABILITY
Q40	0.0001 (1 in 10000)
Q30	0.001 (1 in 1000)
Q20	0.01 (1 in 100)
Q10	0.1 (1 in 10)

SEQUENCE IDENTIFIER

SEQUENCE

SEPARATOR

QUALITY SCORES

Each basecall → quality predictor values

Quality table / Quality list → lists combination of quality predictor scores

Q SCORE BINS	Example of empirically mapped Q scores
N (no call)	N (no call)
2-9	6
10-19	15
20-24	22
25-29	27

Quality score binning

ONT SEQUENCING – OUTPUT FILES

Basecalling → process of converting the electrical signals generated by a DNA strand passing through the nanopore → corresponding base sequence of the strand

Raw data → changes in ionic current → measured by MiniKNOW software

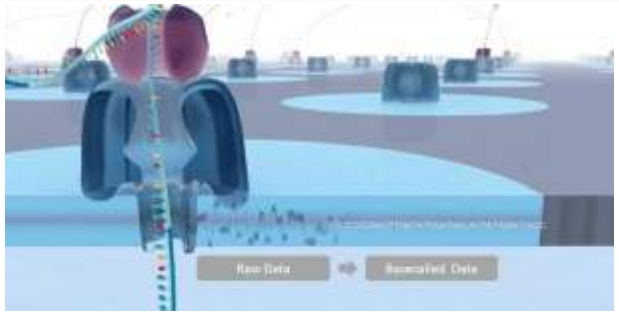
"Signals" → "reads" → POD5 output

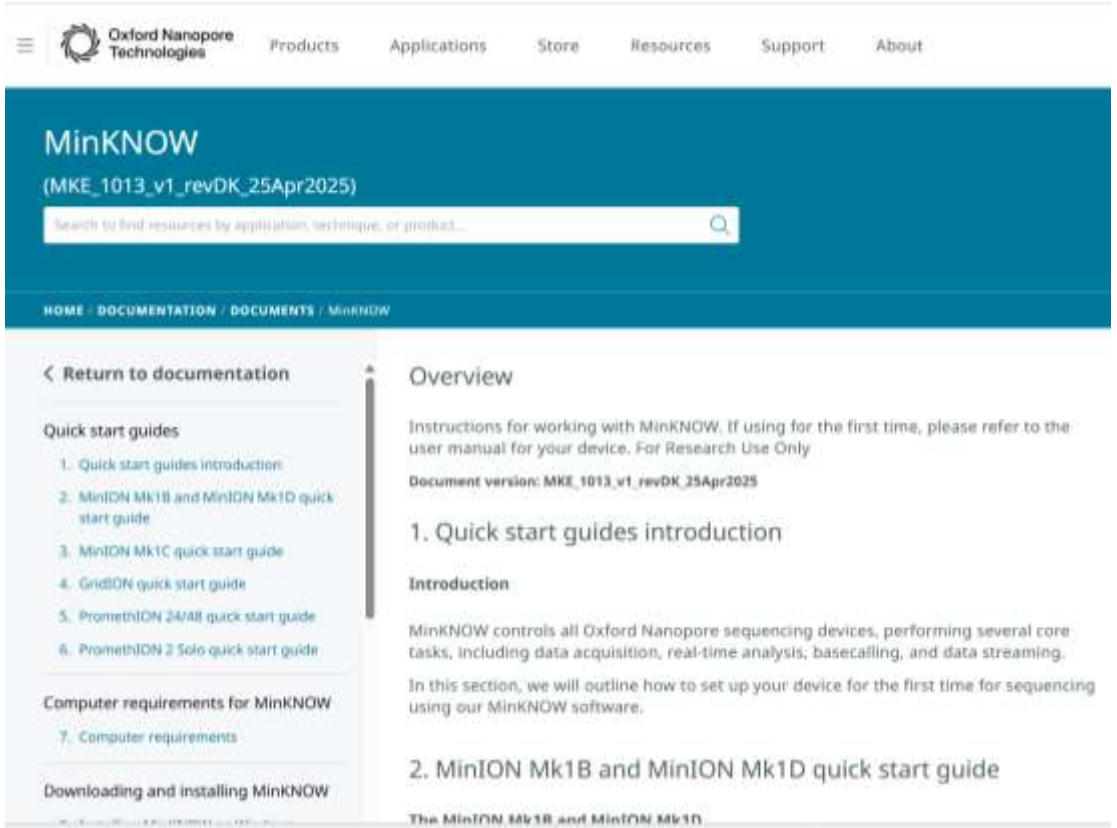
FASTQ files also produced

Basecalling algorithm → 4000 reads / file

ONT Basecallers → MiniKNOW, Dorado, Research algorithms

Output files → POD5 , FAST5 , FASTQ





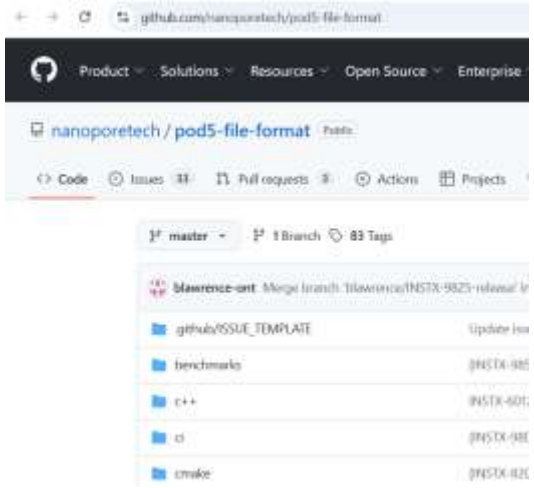
https://nanoporetech.com/document/experiment-companion-minknow

ONT SEQUENCING – OUTPUT FILES

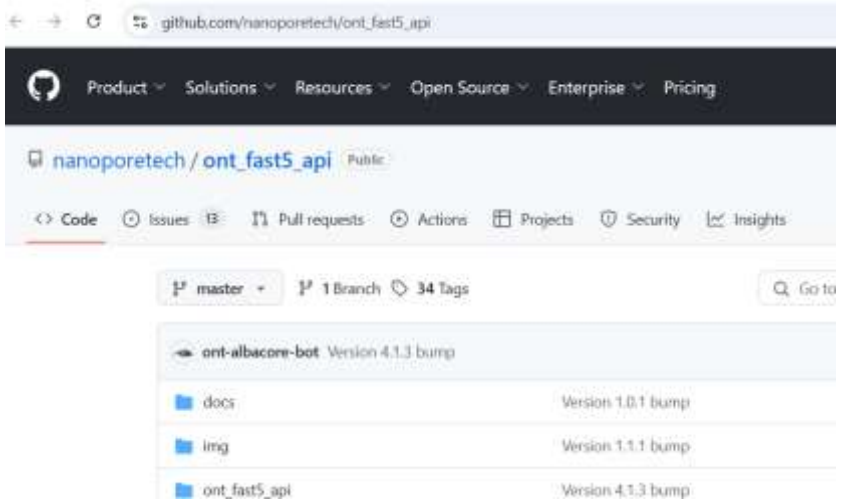
POD5 files

- **More accessible**
- **Reads and writes faster**
- **Occupies less space**
- **Data stored using *Apache Arrow***

- **Fast5 - implementation of HDF5 file format, with specific data schemas for ONT sequencing data**
- **HDF5 file format - a portable file format for storing and managing data**
- **FASTQ files**



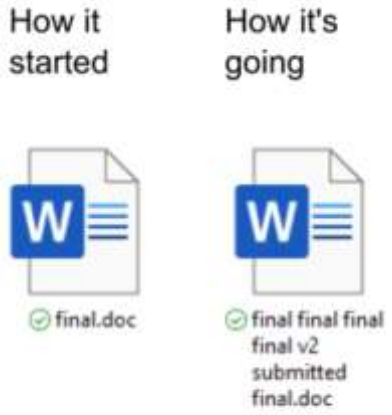
https://github.com/nanoporetech/pod5-file-format

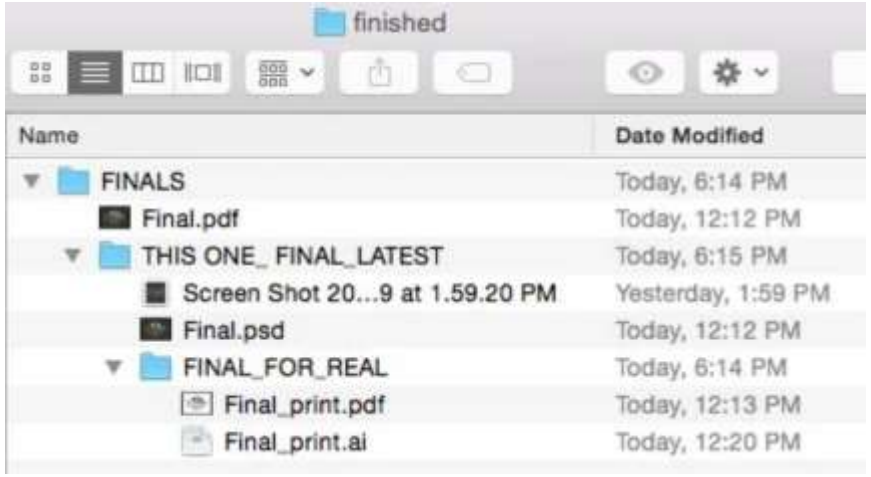


https://github.com/nanoporetech/ont_fast5_api

FILE NAMING GUIDELINES

- IDENTIFY YOUR FILE TYPE
- USE VERSIONING
- SORTING AND RETRIEVING FILES
- FILE NAMES
- CONFIDENTIALITY AND SECURITY





FILE NAMING GUIDELINES

- **IDENTIFY YOUR FILE TYPE**
- USE VERSIONING
- SORTING AND RETRIEVING FILES
- FILE NAMES
- CONFIDENTIALITY AND SECURITY

Identify multiple versions of the same file

EXAMPLE:

Bacteria_203.fasta

salmonella_22.fastq

samplefromasia.fasta

EXAMPLE:

salmonella_bhutan_20190911_024.fastq
salmonella_bhutan_20190911_025.fastq
ecoli_asia_0022.fasta
ecoli_asia_0228.fasta

FILE NAMING GUIDELINES

IDENTIFY YOUR FILE TYPE

USE VERSIONING

SORTING AND RETRIEVING FILES

FILE NAMES

CONFIDENTIALITY AND SECURITY

- Identifying sample type
- Identifying sample origin
- Identifying sample ID
- identifying duplicates

EXAMPLE:

Salmonella_Illumina_batch_final.sh
Salmonella_Illumina_batch_final_FINAL.sh
Salmonella_ente_Illumina_batch_final.sh

EXAMPLE:

salmonella_illumina_batch_v01.sh
salmonella_illumina_batch_v02.sh
salmonella_illumina_batch_v01_01.sh

FILE NAMING GUIDELINES

IDENTIFY YOUR FILE TYPE

USE VERSIONING

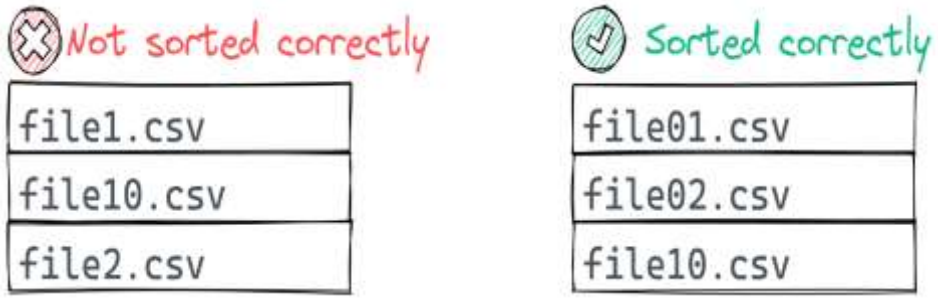
SORTING AND RETRIEVING FILES

FILE NAMES

- date , isolate ID , sample name , sequencing method
- sequencing method, sample name , isolate ID , date

EXAMPLE:

salmonella_2017.fasta
salmonella_20171.fasta
illumina_salmonella_2017.fasta



EXAMPLE:

salmonella_illumina_20171203_0022.fasta
salmonella_illumina_20171203_0023.fasta

20171203_salmonella_illumina_001.fasta
20171203_salmonella_illumina_002.fasta
20171203_salmonella_illumina_003.fasta
20171203_ecoli_illumina_001.fasta
20171203_ecoli_illumina_002.fasta

FILE NAMING GUIDELINES

IDENTIFY YOUR FILE TYPE

USE VERSIONING

SORTING AND RETRIEVING FILES

FILE NAMES

CONFIDENTIALITY AND SECURITY

- Avoid vague words
- Avoid full stops
- Avoid special characters
- Avoid long names
- Maybe abbreviations
- Follow consistent date format (YYYYMMDD)

EXAMPLE:

salmonella_sample_first.fasta
salmonella_sample_second_last.fasta
salmonella_sample_last.fasta
salmonella_sample_last_LAST.fasta

EXAMPLE:

salmonella_india_20231212_001.fasta
salmonella_india_20231212_098.fasta
salmonella_india_20231212_099.fasta
salmonella_india_20231212_100.fasta

HASHING

2cf24dba5fb0a30e26e83b2ac5b

MD5 HASH

DATA + HASHING FUNCTION = HASH

Cannot get original data from a HASH

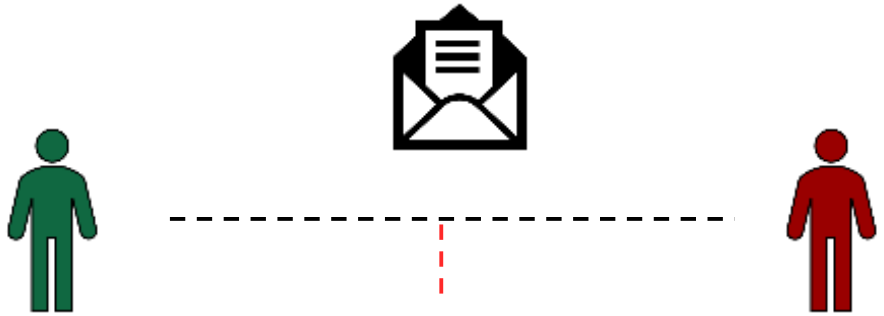
Same DATA = Same HASH ; Different data = Different Hash

verified by using HASH

Has the HASH been altered?

https://www.md5hashgenerator.com/

CHECKSUMS



1001
0101
0123

10

1001
5555
0123

10

≠

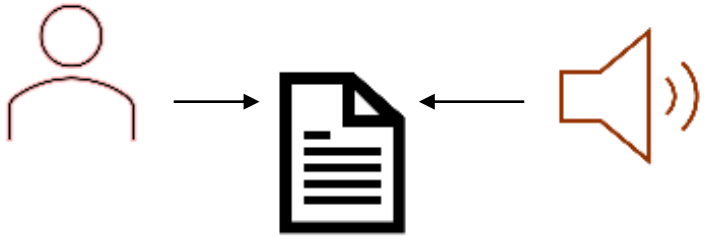
28

1001
5555
0123

28

10

CHECKSUM → complex cryptographic hash functions



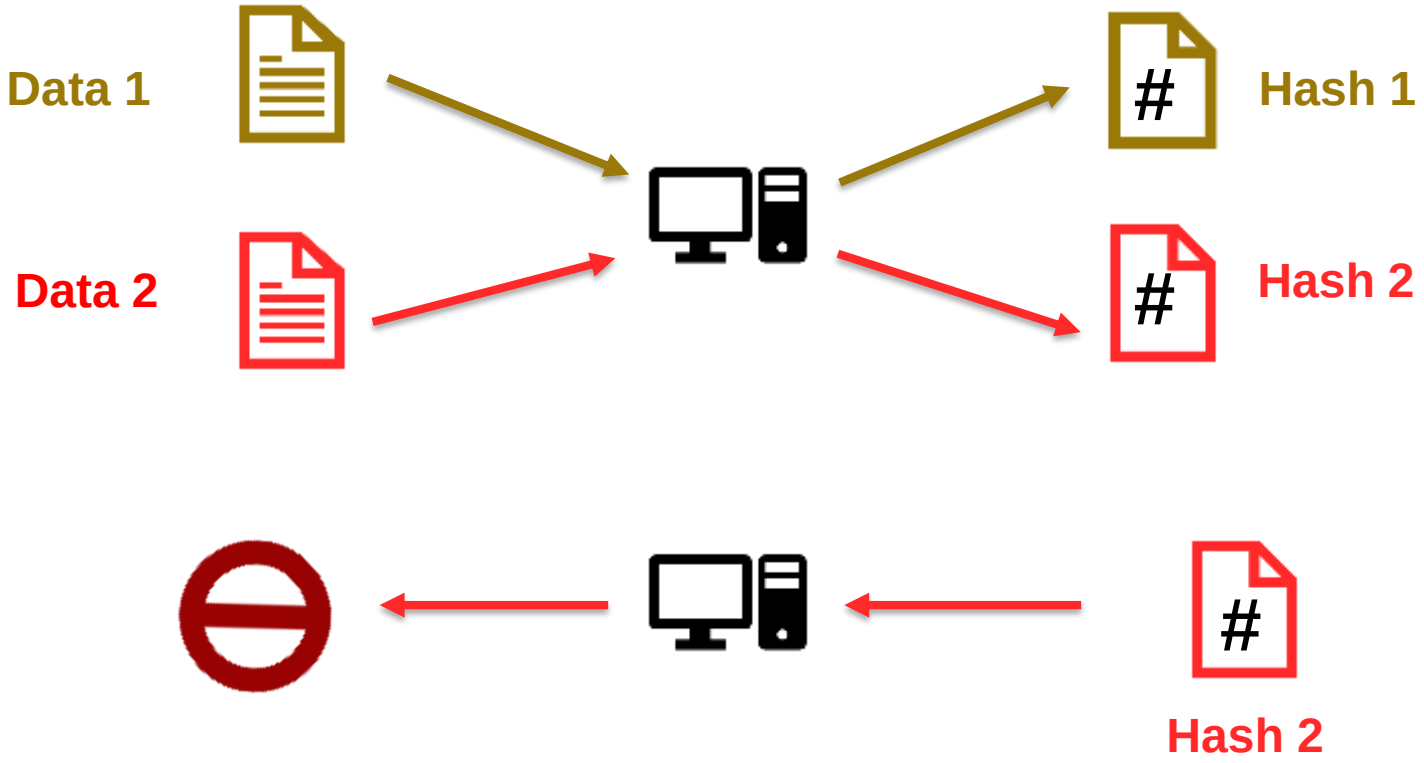
INPUT

WEBSITE

OUTPUT

SHA256

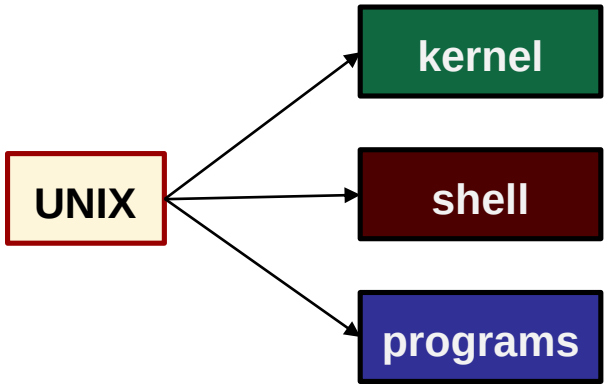
- SHA-256 → Secure Hash Algorithm 256-bit
- Outputs a fixed 256-bit (64 hexadecimal characters) hash from any input.
- Input data → passed through a cryptographic hash function → outputs a unique hash
- Even a 1-bit change in input = a completely different hash
- same input = same hash
- 4 bits = 1 character



SHA family	Bit size	Hash characters
SHA - 1	160 bits	40 characters
SHA - 224	224 bits	56 characters
SHA - 256	256 bits	64 characters
SHA - 384	384 bits	96 characters
SHA - 512	512 bits	128 characters

LINUX / UNIX operating systems

- the suite of programs which make the computer work
- It is stable, multi-user, multi-tasking system for servers, desktops and laptops



allocates time & memory, handles the filestore and communications

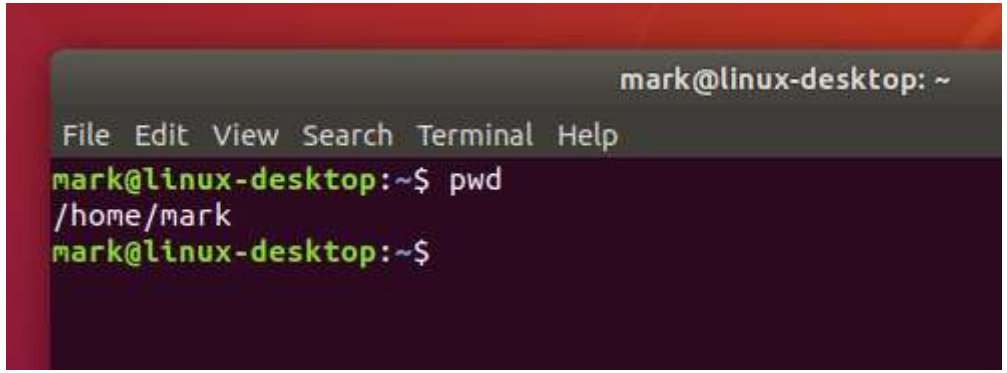
interface between user and kernel → command line interpreter (CLI)

core set of utility programs to carry out actions like file management

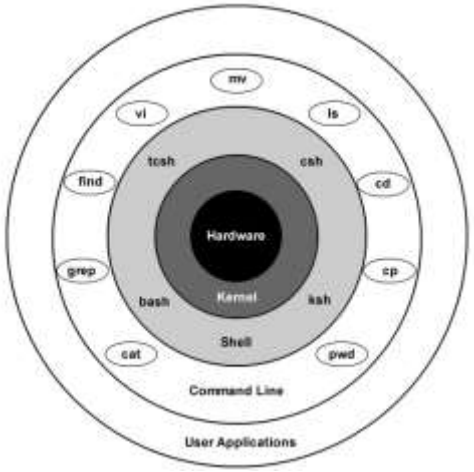
LINUX vs UNIX → share the same idea of the shell and kernel

UNIX → used for high end operations

LINUX → easily downloadable and operable



UNIX COMMANDS



Allows users to customize shell environment and write own shell scripts

Usage of pipes (|) → links workflows unidirectionally

Eg: ls | grep file.txt

```
root@gfgubun1:/home/jayesh# ls
 Desktop      file.txt                    Pictures    Templates
 Documents   'hello jayesh this side'     Public      Videos
 Downloads    Music                      snap
root@gfgubun1:/home/jayesh# ls | grep file.txt
file.txt
root@gfgubun1:/home/jayesh#
```

More commands …..

https://www.w3schools.com/bash/

Linux / Unix TUTORIAL

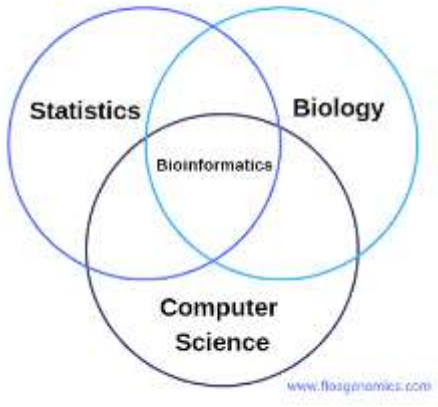
JupyterHub

BIOINFORMATICS

Using computer technology to collect, store, analyze and disseminate biological data

GOALS:

- **Development of efficient algorithms**
- **Extension of experimental data by predictions**
- **Increase understanding of biological processes**
- **Interpretation of biological data to correlate results**
- **Solve practical problems in data storage, management and sharing**



DATA IN BIOINFORMATICS

- **Classic data: DNA sequences of genes, full genomes , amino acid sequence of proteins**
- **“omics” data: transcriptomics, proteomics, interactomics, metabolomics**
- **Metagenomics and Metaproteomics**

COMPUTATIONAL COMPONENTS

HARDWARE

High-performance computing (HPC) clusters

Cloud-based servers → AWS, Google Cloud, Azure

Local machines → workstations or laptops

OPERATING SYSTEMS

Windows, Mac, Linux, UNIX

SOFTWARE TOOLS

Sequence alignment tools → BLAST, BWA

Data visualization tools → IGV, UCSC Genome Browser

Statistical tools → R, Bioconductor

PROGRAMMING LANGUAGES

Python, Perl, R, Bash

BIOCONDA

PACKAGES

Conda, Bioconda, ….

DATABASES

NCBI, UniProt, EMBL,

WORKFLOW MANAGEMENT SYSTEMS

Nextflow, Snakemake, Galaxy

! COMMON ISSUES !

i. Dependency conflicts
ii. Installation issues
iii. Reproducibility concerns across systems

SOLUTION
Docker containers, Virtual Machines
Or using environments !

Developing Oracle weblogic software

Testing out the software

Deploying the software

Installation – 3x !!!!

DOCKER CONTAINERS

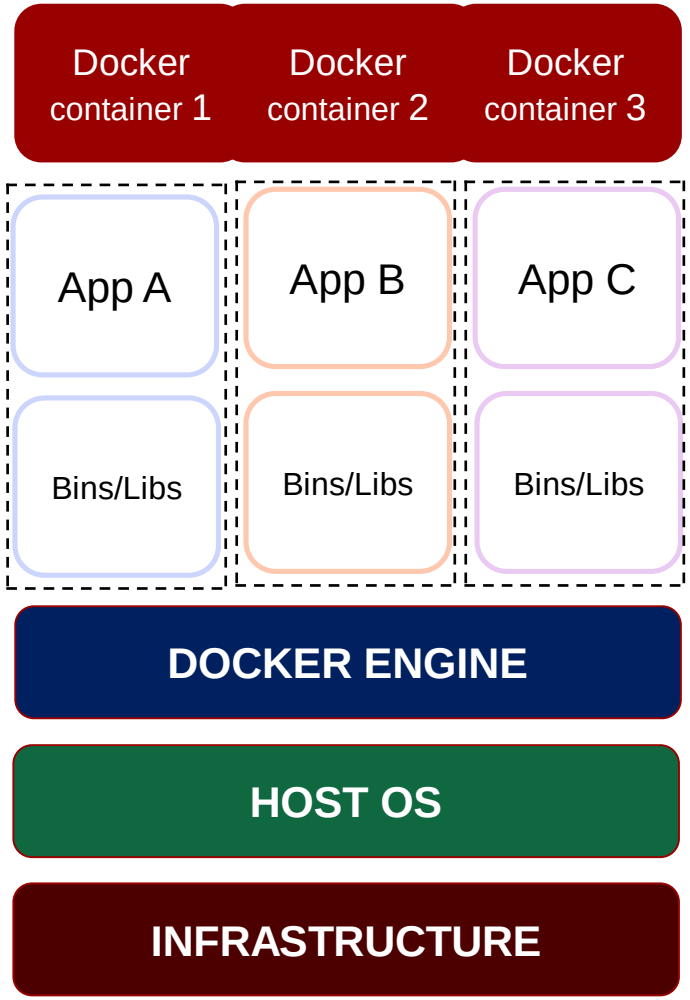
- **Open-source containerization software platform**
- **Create deploy and manage applications in containers**
- **Applications + environment → provided in parallel**

Steps:

i. **Develop application and supporting components using containers**

ii. **Container → unit for distributing and testing out application**

iii. **Deploy application into production environment (local data center / a cloud provider / hybrid)**



CONTAINERS

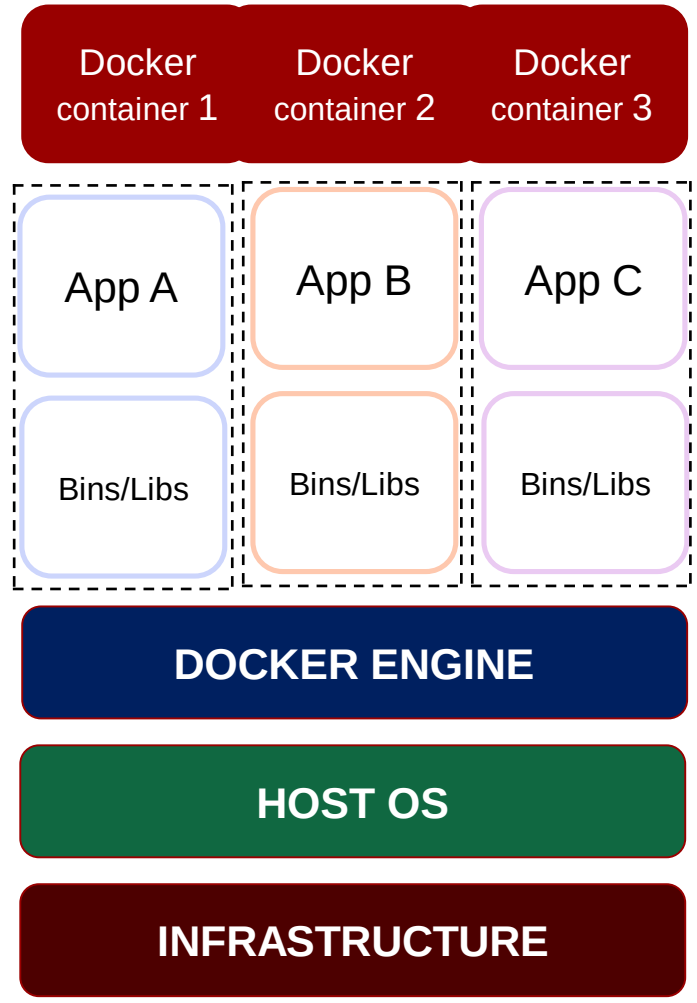
- **Containers help in creating an isolated environment → “sandbox”**
- **Virtualize the host OS and isolate an applications dependencies from other containers running on the same machine**
- **Uses → portability, lightweight, less disk space, secure, speed**
- **Container engine + container image → package of application + its dependencies**

Advantage

✓No need to run separate operating system / application

✓ Lower costs

✓Higher resource utilization



VIRTUAL MACHINES

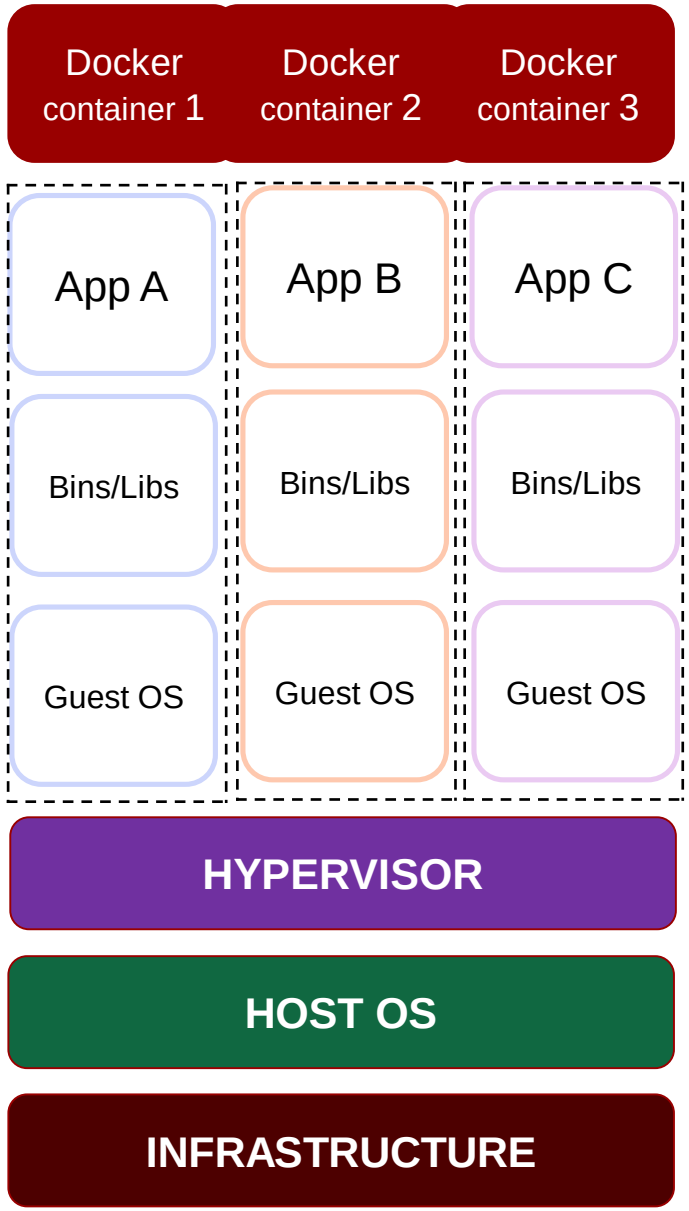
- **VM → individual computer inside a host machine (computer)**
- **Multiple VMs can live inside a single host machine**
- **Uses host machine's hardware**
- **Hypervisor → piece of software that virtualizes the host's hardware and manages feeding resources to the VMs**

Disadvantages:

✓ **computational overhead**

✓ **Own OS installed → requires several Gb of storage**

✓ **Takes time to install → has to be done each time**



COMPUTING ENVIRONMENTS IN BIOINFORMATICS

environments make it easy to install a wide variety of command-line tools → prevents them from interfering with one another

Main framework

Speeds up conda installations

Additional packages in Bioinformatics

COMPUTING ENVIRONMENTS IN BIOINFORMATICS

environments make it easy to install a wide variety of command-line tools → prevents them from interfering with one another

CONDA

Main framework

- **Create environments**
- **Listing environments**
- **Installing packages**
- **Specifying channels**

Mamba

Speeds up conda installations

```
conda create -n <env-name>
```

```
conda info --envs
```

```
# via environment activation
conda activate myenvironment
conda install matplotlib
```

```
conda install conda-forge::numpy
```

BIOCONDA

Additional packages in Bioinformatics

```
conda create -n myenvironment python numpy pandas
```

```
conda environments:

   base           /home/username/Anaconda3
   myenvironment   * /home/username/Anaconda3/envs/myenvironment
```

```
# via command line option
conda install --name myenvironment matplotlib
```

https://docs.conda.io/projects/conda/en/stable/user-guide/getting-started.html

COMPUTING ENVIRONMENTS IN BIOINFORMATICS

environments make it easy to install a wide variety of command-line tools → prevents them from interfering with one another

Main framework

Speeds up conda installations

Additional packages in Bioinformatics

- **Create environments**
- **Listing environments and installing packages**

```
mamba create -n nameofmyenv <list of packages>
```

`mamba install`

```
mamba create -n myjlabenv jupyterlab -c conda-forge
mamba activate myjlabenv  # activate our environment
jupyter lab               # this will start up jupyter lab and open a browser
```

https://mamba.readthedocs.io/en/latest/user_guide/mamba.html

COMPUTING ENVIRONMENTS IN BIOINFORMATICS

environments make it easy to install a wide variety of command-line tools → prevents them from interfering with one another

CONDA

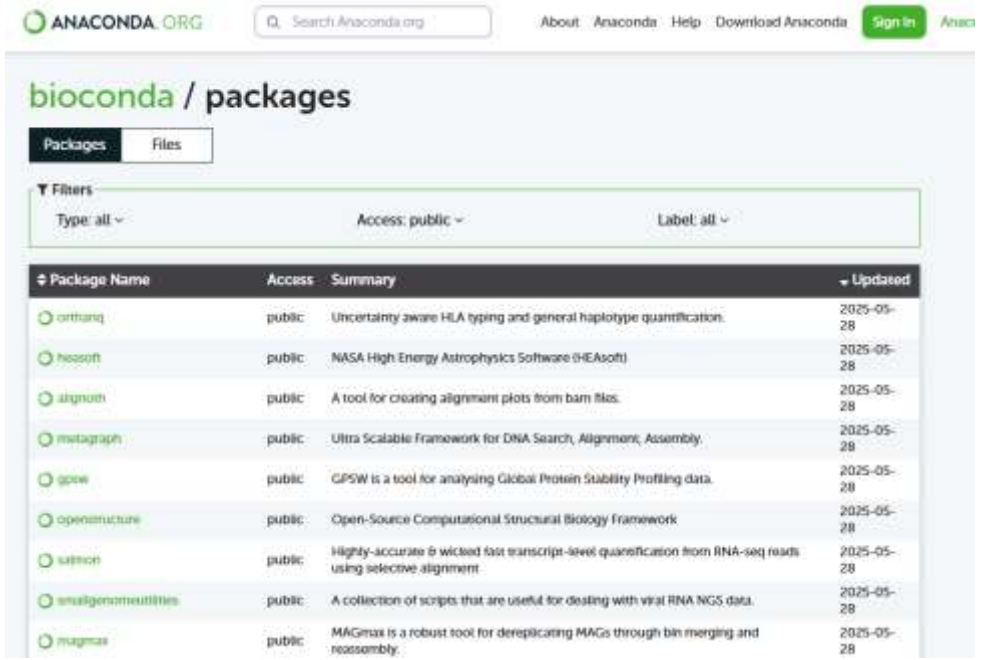
Main framework

Mamba

Speeds up conda installations

BIOCONDA

Additional packages in Bioinformatics



- **Install conda first**
- **always cite it**

```
conda config --add channels bioconda
conda config --add channels conda-forge
conda config --set channel_priority strict
```

https://bioconda.github.io/

THANK YOU FOR LISTENING, ANY QUESTIONS ?

Kindly ensure to drop your email address on the chat to receive the certificates