BIOINFORMATICS ESSENTIALS

PRESENTED BY: Praissy Zefi J (DTU)
Bioinformatician
pzeje@dtu.dk

LEARNING OBJECTIVES

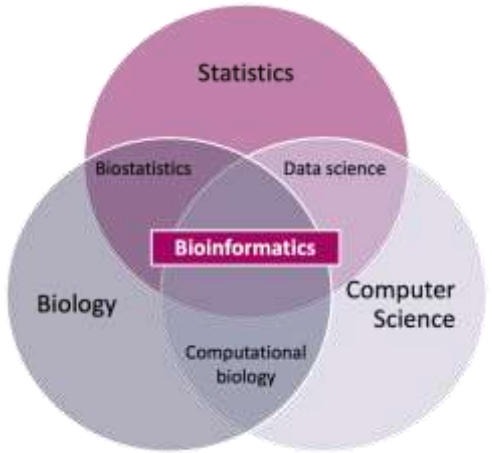
- Introduction to Bioinformatics in Biology
- Environments to use Bioinformatics tools
- Pipelines to streamline bioinformatic processes
- Storing, managing and sharing Biological data
- Validation of Bioinformatics results

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

BIOCONDA

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
PACKAGES	Conda, Bioconda, ….
DATABASES	NCBI, UniProt, EMBL,
WORKFLOW MANAGEMENT SYSTEMS	Nextflow, Snakemake, Galaxy

nextflow

Galaxy PROJECT

COMPUTATIONAL COMPONENTS

HARDWARE

High-performance computing (HPC) clusters
Cloud-based servers → AWS, Google Cloud, Azure
Local machines → workstations or laptops

, Mac, Linux, UNIX

nent tools → BLAST, BWA
s → IGV, UCSC Genome Browser
ols → R, Bioconductor

PACKAGES

Python, Perl, R, Bash

da, Bioconda, ….

WORKF

BI, UniProt, EMBL,

Nextflow, Snakemake, Galaxy

! COMMON ISSUES !

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

SOLUTION
using environments !

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

Main framework

Speeds up conda installations

Additional packages in Bioinformatics

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

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

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

```
conda info --envs
```

```
conda environments:

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

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

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

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

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

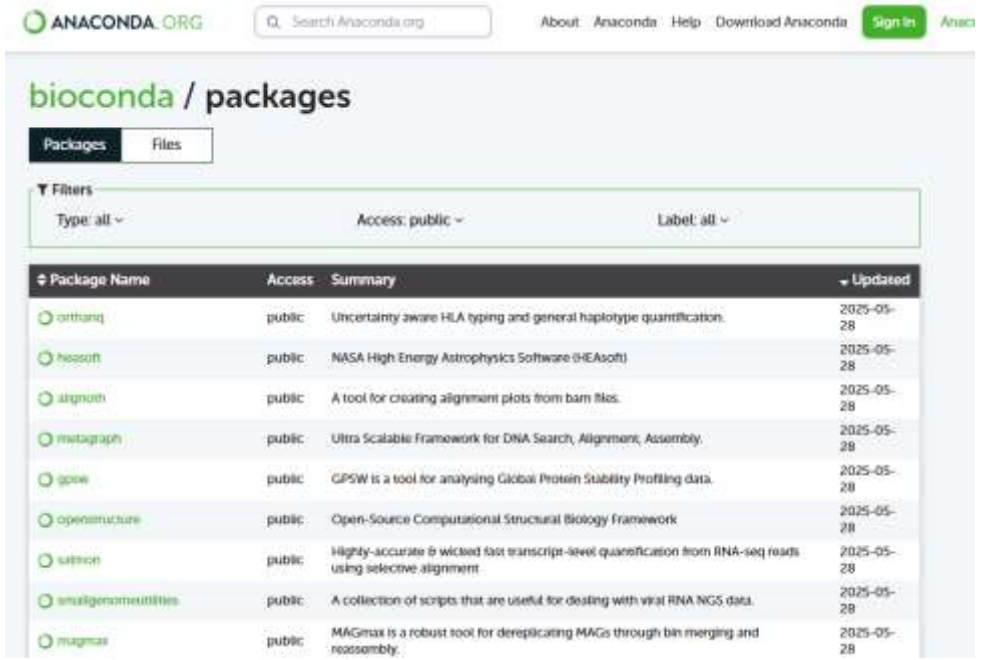
Main framework

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/

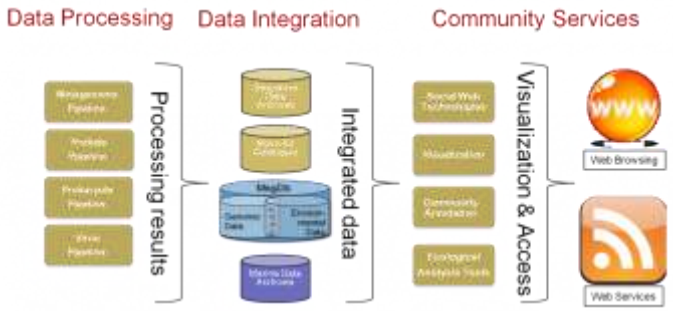
CONDA , MAMBA, BIOCONDA

TUTORIAL

DATA SHARING IN BIOINFORMATICS: COMMON GUIDELINES

DATA and METADATA

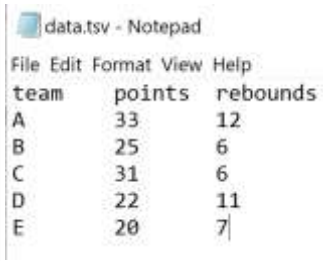
- Data → recorded observations ; METAData → description of primary data and resources to get it
- MUST share metadata while sharing data
- Describes important protocol and bioinformatic processes



DOCUMENTATION

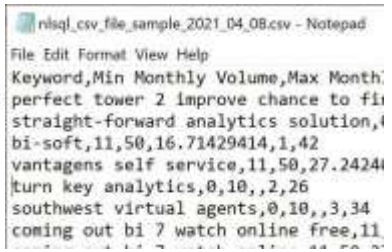
README files → file organization, location, observations and variables present

1. Source code → preprocessing steps and analysis scripts
2. Licensing → copyright laws, etc.


data.tsv - Notepad

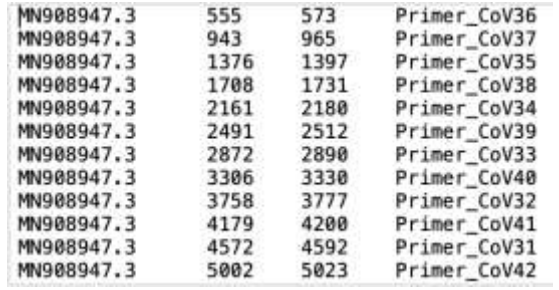
File Edit Format View Help

team	points	rebounds
A	33	12
B	25	6
C	31	6
D	22	11
E	20	7


nlsql_csv_file_sample_2021_04_08.csv - Notepad

File Edit Format View Help

```
Keyword,Min Monthly Volume,Max Month
perfect tower 2 improve chance to fi
straight-forward analytics solution,
bi-soft,11,50,16.71429414,1,42
vantagens self service,11,50,27.2424
turn key analytics,0,10,,2,26
southwest virtual agents,0,10,,3,34
coming out bi 7 watch online free,11
```



MN908947.3	555	573	Primer_CoV36
MN908947.3	943	965	Primer_CoV37
MN908947.3	1376	1397	Primer_CoV35
MN908947.3	1708	1731	Primer_CoV38
MN908947.3	2161	2180	Primer_CoV34
MN908947.3	2491	2512	Primer_CoV39
MN908947.3	2872	2890	Primer_CoV33
MN908947.3	3306	3330	Primer_CoV40
MN908947.3	3758	3777	Primer_CoV32
MN908947.3	4179	4200	Primer_CoV41
MN908947.3	4572	4592	Primer_CoV31
MN908947.3	5002	5023	Primer_CoV42

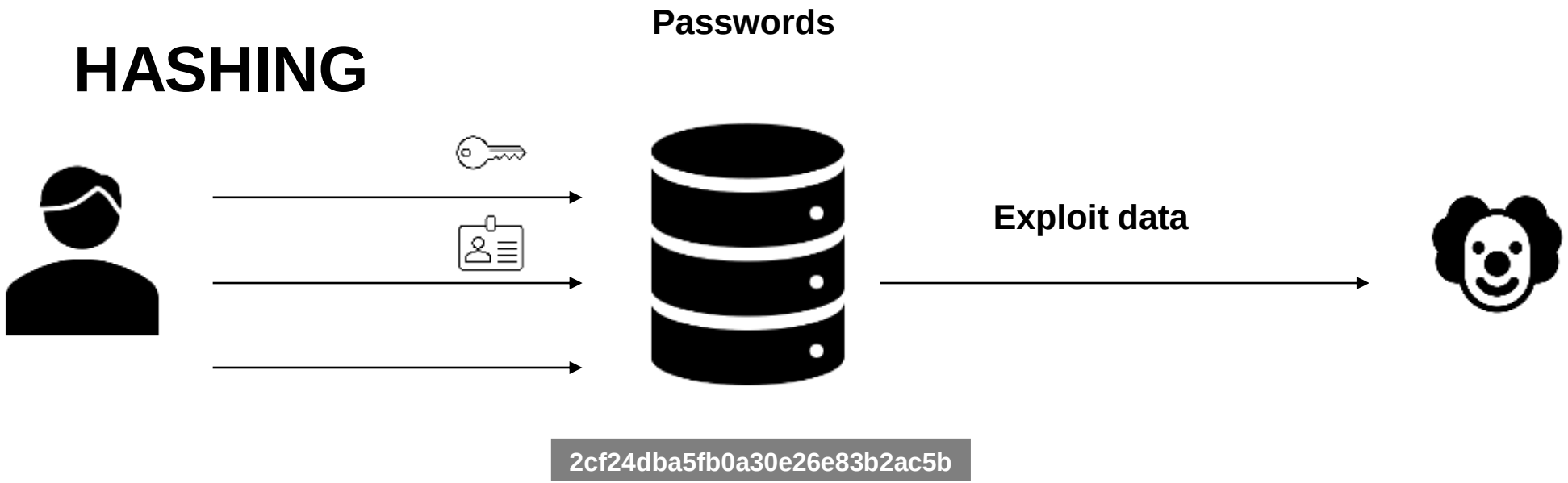
SHARING DATA

File format: tabular data → .tsv & .xlsx ; .csv – do not!

Compressing files → .fasta & .fastq → .CRAM ; BAM (binary alignment/map) , .gzip files → single files ; .zip → collection of files

Genomic regions → .bed files (browser extensible data)

HASHING



2cf24dba5fb0a30e26e83b2ac5b

MD5 HASH

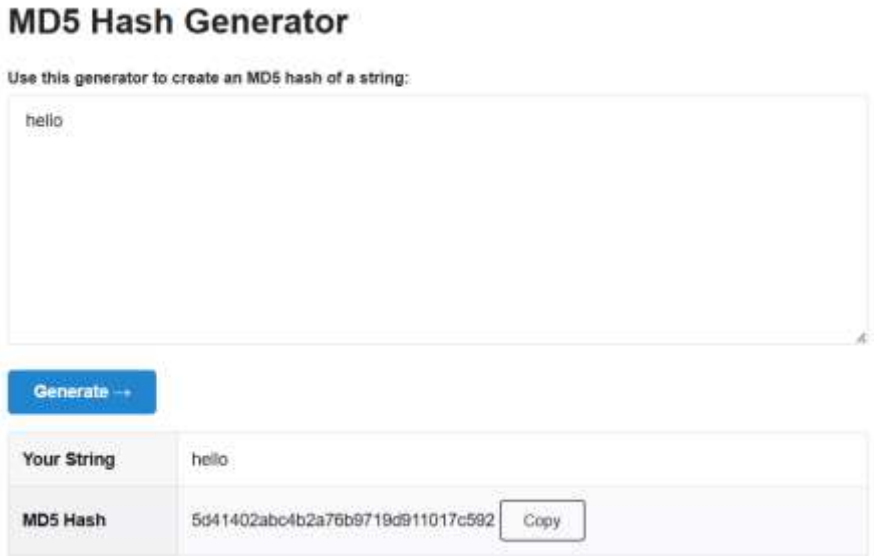
DATA + HASHING FUNCTION = HASH

Cannot get original data from a HASH

Same DATA = Same HASH ; Different data = Different Hash

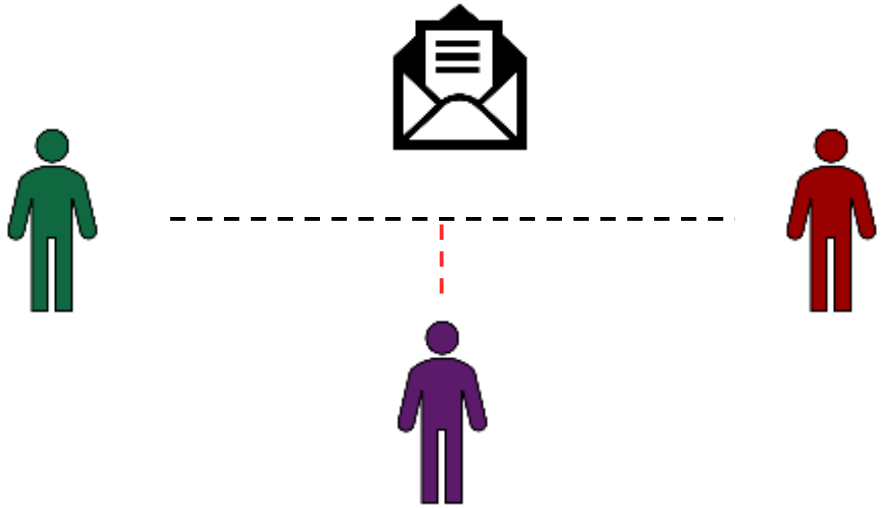
verified by using HASH

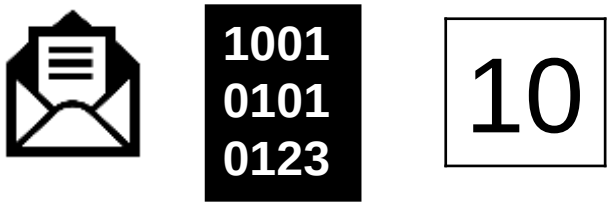
Has the HASH been altered?

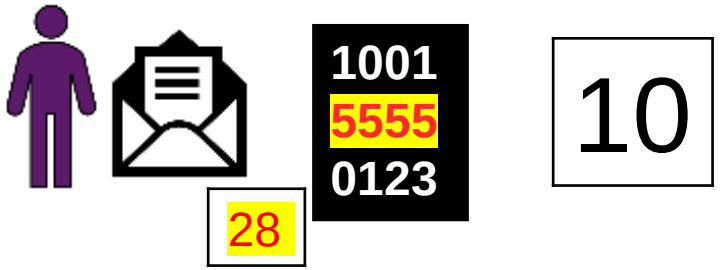


https://www.md5hashgenerator.com/

CHECKSUMS



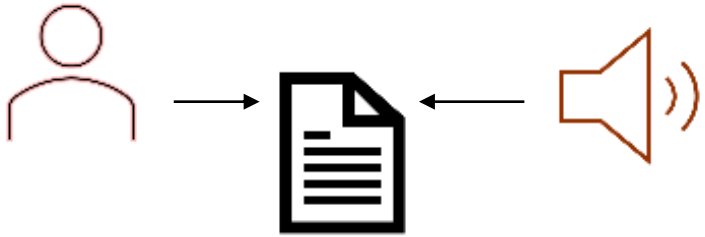




≠

28

CHECKSUM → complex cryptographic hash functions



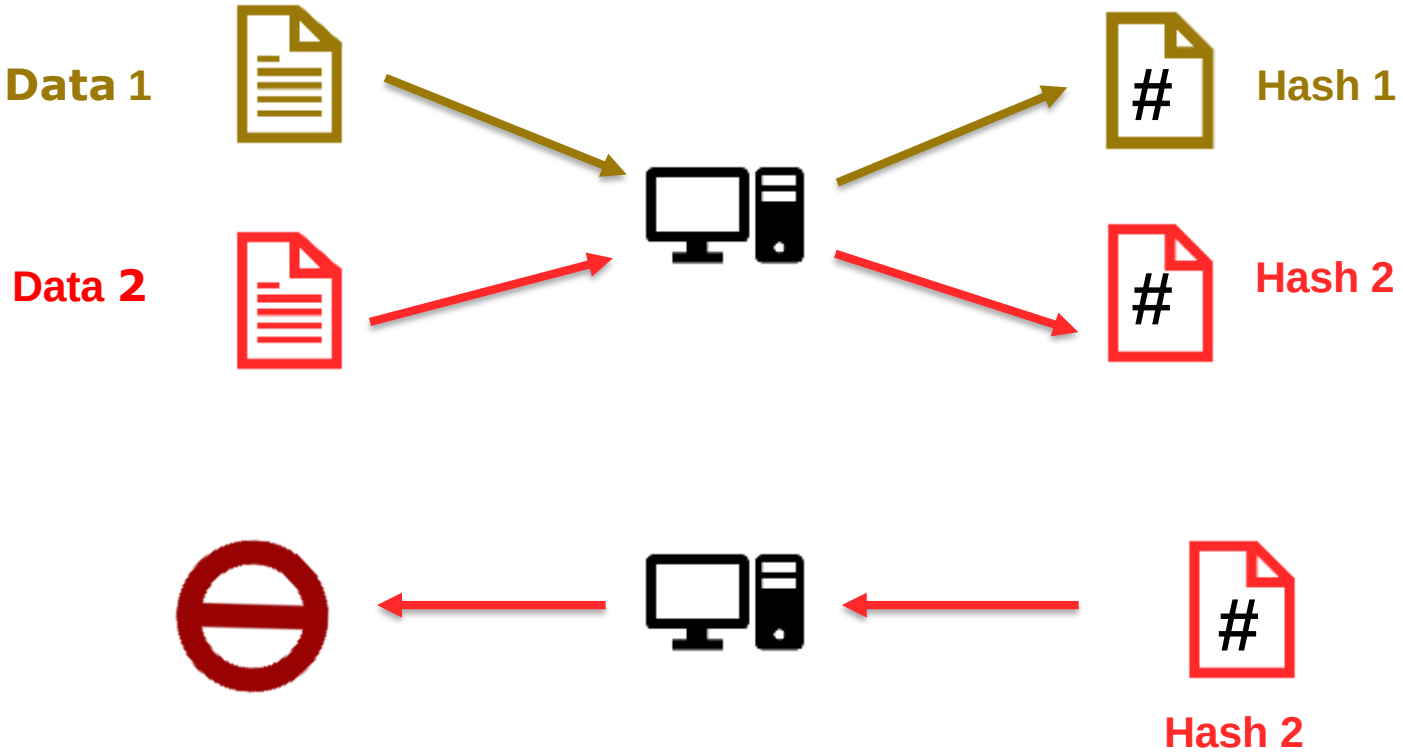
INPUT

WEBSITE

OUTPUT

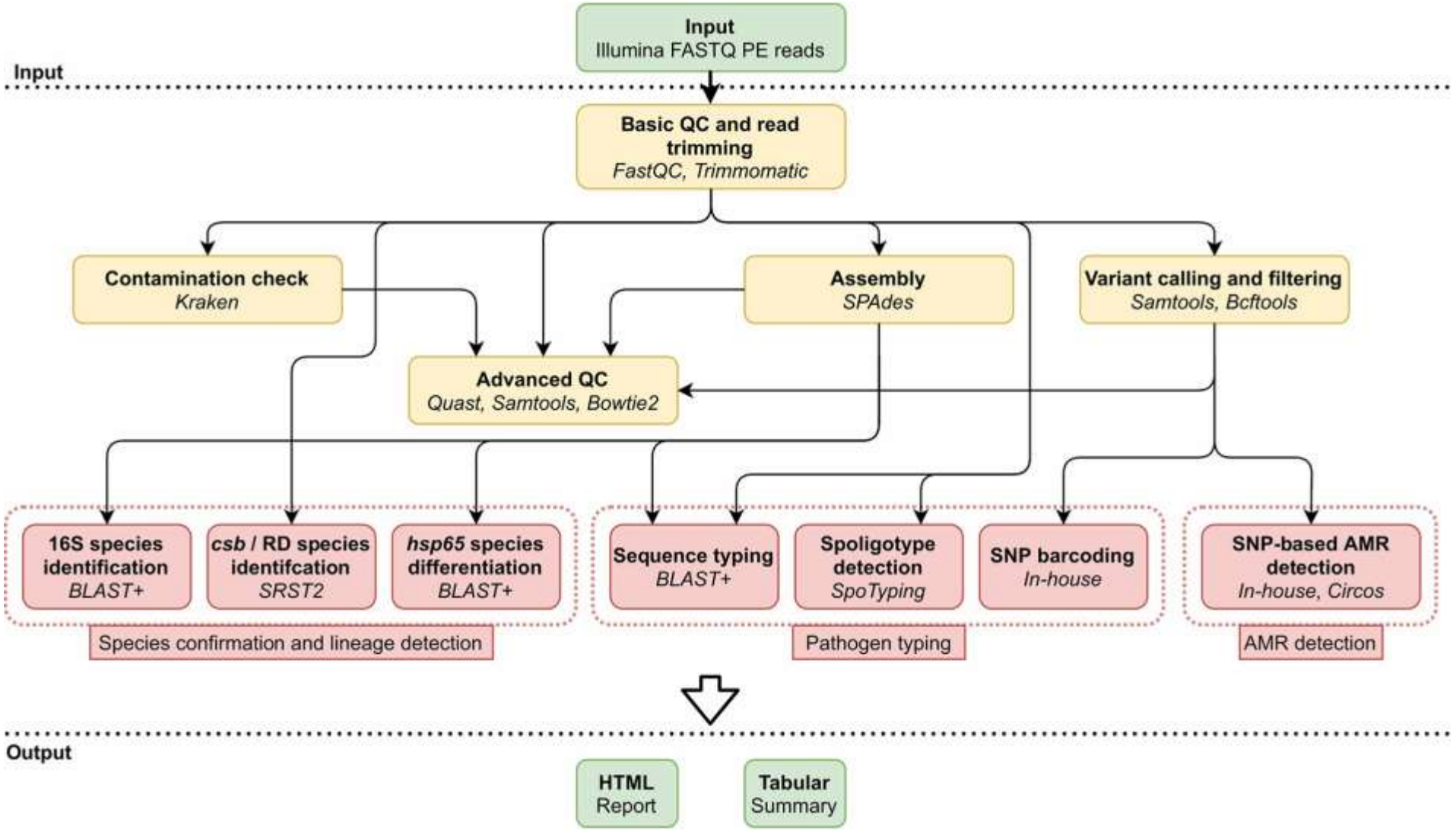
SHA256

- 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

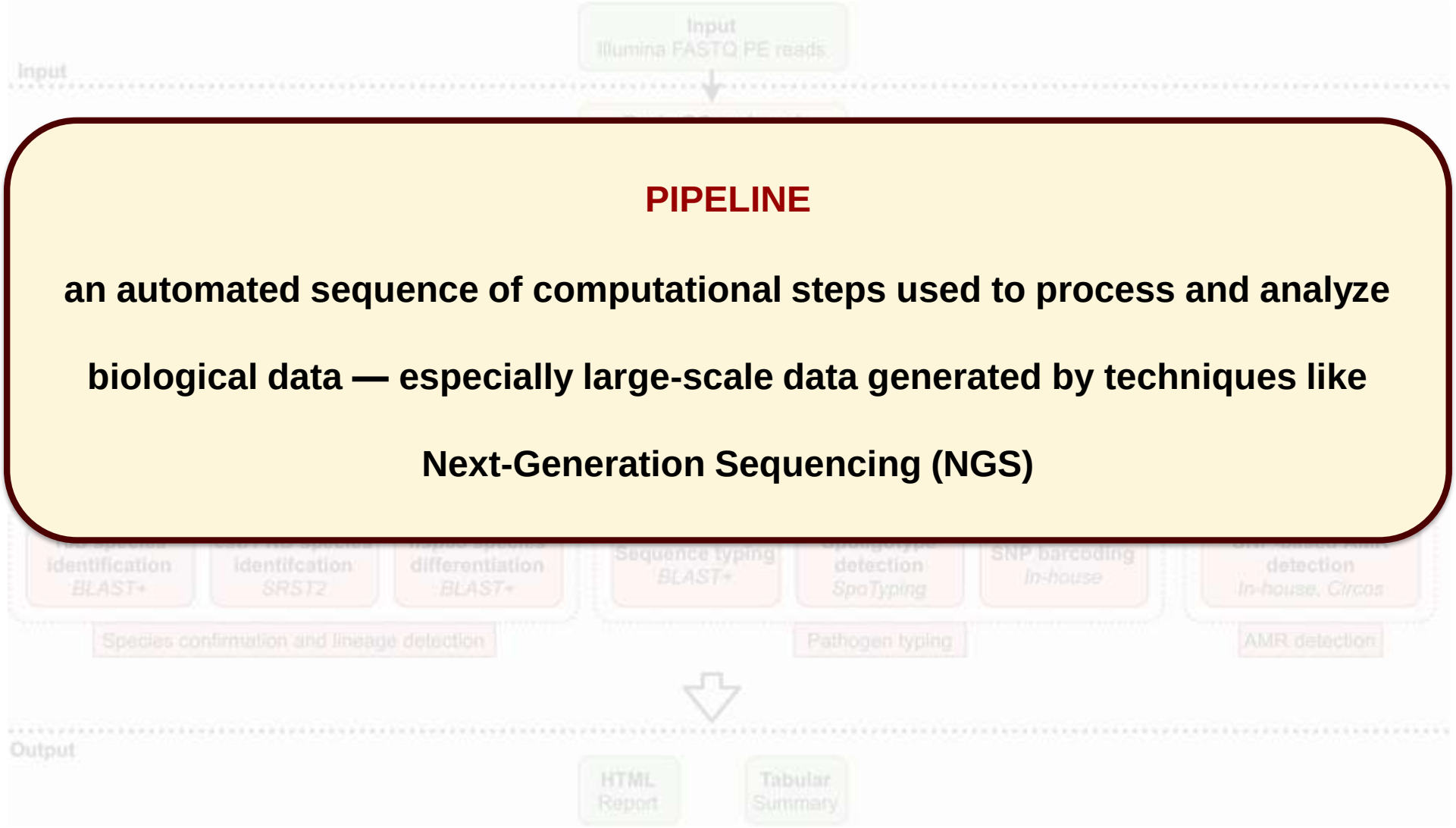


SHA family	Bit size	Hash characters
SHA - 1	160 bits	40 characters
SHA - 224	224 bits	56 characters
SHA - 256	256 bits	64 characters
SHA - 384	384 bits	96 characters
SHA - 512	512 bits	128 characters

BIOINFORMATICS PIPELINES



BIOINFORMATICS PIPELINES



PIPELINE

an automated sequence of computational steps used to process and analyze biological data — especially large-scale data generated by techniques like Next-Generation Sequencing (NGS)

PLANNING A PIPELINE

1 – Plan	Trimmomatic, fastQC, CutAdapt
2 – Break into smaller workflows	Data cleaning script, alignment script, etc
3 – Combine scripts into a single pipelines	Raw reads → input of 2nd script → input to 3rd
4 – Handle dependencies	Workflow managers!
5 – Automate the pipeline	
6 – Error handling and logging	**Eg:** fastqc *.fastq.gz > logs/fastqc.log 2>&1
7 – Version control and documentation	Use the correct versions of tools to install
8 – Scaling up	snakemake --jobs 20 --cluster "sbatch -A project --time=2:00:00"

PLANNING A WORKFLOW

1 – Plan

2 – Break into smaller workflows

3 – Combine scripts into a single pipelines

6 – Error handling and logging

7 – Version control and documentation

8 – Scaling up

Trimmomatic, fastQC, CutAdapt

snakemake --jobs 20 --cluster "sbatch -A project --time=2:00:00"

! LIMITATIONS !

Large volume of data and complex processes

WORKFLOW MANAGEMENT SYSTEMS

I. Streamlines process
II. “modular”
III. Pipeline built by blocks
IV. Saves time
V. Ensure consistency

PLANNING A WORKFLOW

nextflow

- Very flexible
- Highly portable

snakemake

- Python based
- Designed for local works on cluster & cloud
- Beginner level (easy syntax)

Galaxy

- Web based
- Great GUI
- No coding knowledge required

WORKFLOW MANAGEMENT SYSTEMS

I. Streamlines process

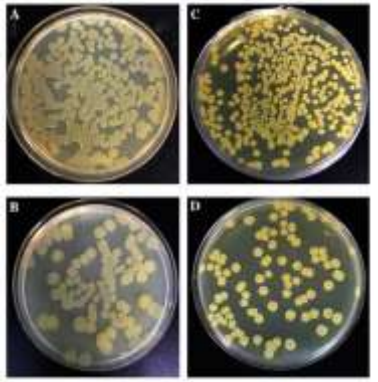
II. “modular”

III. Pipeline built by blocks

IV. Saves time

V. Ensure consistency

VALIDATION OF BIOINFORMATICS RESULTS



GENOTYPE → Complete set of genes that an organism carries

PHENOTYPE → observable characteristics influenced by genotype and environment

Validating predicted genotype to phenotype link ensures accuracy, reliability and reproducibility of results

EXAMPLE:

E.coli isolate carries the blaCTX-M gene (a gene encoding an extended-spectrum beta-lactamase), and laboratory testing shows that the isolate is resistant to cefotaxime (a third-generation cephalosporin), this indicates a good correlation between the genotype (blaCTX-M) and the phenotype (cefotaxime resistance)

BREAKDOWN:

Bacterial isolate with blaCTX-M gene → resistant to cefotaxime

Test bacteria in the lab when exposed to cefotaxime → grows well

Cefotaxime resistant !

Genotype: Has the blaTEM gene ; Phenotype: Is resistant to ampicillin

WGS enables rapid prediction of antimicrobial resistance (AMR).

Tools like ResFinder, AMRFinderPlus, CARD predict resistance from DNA sequences.

But genotype ≠ phenotype — presence of gene ≠ resistance expression.

Genotype = Blueprint and Phenotype = Functional Outcome

GENOTYPE vs PHENOTYPE

A bacterial genotype can correlate with its resistance phenotype (which is the observable effect of resistance to antibiotics)

Depends on 1) bacterial species 2) the specific antibiotic 3) the presence of other factors like regulatory mechanisms or mutations.

VALIDATING RESULTS

I.	***in-silico* validation (computer-based)**	**CARD, AMRFinder Plus, PATRIC, VFDB**
II.	**Phenotype validation (wet-lab)**	**Antimicrobial susceptibility testing (AST)**
III.	**Cross-validation with external databases**	**Using previously validated genomes with known phenotypes**

Gene / Mutation	Drug Class Affected	Phenotype Correlation	Notes
mecA	Beta-lactams	High	MRSA detection
blaCTX-M	3rd-gen cephalosporins	High	Common ESBL gene
vanA / vanB	Glycopeptides (vancomycin)	High	VRE hallmark genes
gyrA/parC	Fluoroquinolones	High	Point mutation-based resistance

GENOTYPE vs PHENOTYPE* EXCEPTIONS!

Not all genotypic variations lead to phenotypic changes

- Truncated/pseudogenes → Gene is disrupted due to premature stop codons, frameshifts, or deletions
- Low/No gene expression → resistance gene is present but not actively transcribed or translated
- Non-functional Point Mutations → Mutation exists in the gene but does not affect resistance.
- Gene duplication → Doesn't always increase resistance
- Resistance requires multiple components

Reason	Example	Phenotypic Result
Truncated/pseudogene	*blaTEM* with frameshift	Susceptible to penicillins
Silent mutation	*gyrA* variant not at key position	Susceptible to fluoroquinolones
Low expression	*tetA* with weak promoter	Susceptible to tetracyclines
Gene on unstable plasmid	*qnrB* lost during culturing	Susceptible to quinolones

THANK YOU FOR LISTENING, ANY QUESTIONS ?

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