Linux

for Bioinformatics

Why to use Linux for Bioinformatics?

- Linux is free
- Most Bioinformatics tools are only available in Linux
- Windows is very slow at processing biological data
- Linux has built-in programming languages
- Creating a biological analysis pipeline can be done easily in Linux

Linux Commans for Bioinformatician

- Navigating the file system
- Locating programs and files:
- Text data manipulation in Linux for Bioinformatics
- Pre-processing biological datasets in Linux

Navigating the File System

- PWD: print working directory

```
pwd
```

- CD: changing directories

```
cd /user/rafiga
```

- MKDIR: making directories

```
mkdir ngs_data_anly
```

- CP & MV: copying and moving files, directories, and data

```
mv text.txt  ngs_data_anly
```

```
cp text.txt  ngs_data_anly
```

- RM: deleting files and directories

```
rm text.txt
rm –r ngs_data_anly
```

Locating Programs and Files

- LS: listing files and directories on Linux

```
ls
```

```
ls -lh
```

- WHICH & WHEREIS: finding installed programs

```
which R
Whereis R
```

- FIND: locating user-created files

```
find /user/rafiga -type f –name “*.txt”
```

Text Data Manipulation in Linux for Bioinformatics

- Cat: visualization and inspection of text data
  - Display file:

    ```
    cat text.txt
    ```

  - Display the number of lines in the file:

    ```
    cat –n text.txt
    ```

  - Concatenating files:

    ```
    cat text1.txt text2.txt > concat.txt
    ```

  - Create a new file:

    ```
    cat > new_file.txt
    ```

  - Append to the excisting file:

    ```
    cat >> new_file.txt.   ( + Crtl+D)
    ```

Text Data Manipulation in Linux for Bioinformatics

- Head: reading a specified number of lines from the top

```
head text.txt
```

```
head –n 3 text.txt
```

- Tail: reading a specified number of lines from the bottom

```
tail text.txt
```

```
tail –n 3 text.txt
```

- Reading log files in real-time:

```
tail –f process.log
```

```
tail –fn20 process.log
```

Text Data Manipulation in Linux for Bioinformatics

- LESS: visualization of textual data

```
less text.txt
```

- STAT: retrieving statistics of files and directories

```
stat text.txt
```

```
less /user/rafiga/
```

Pre-processing Biological Datasets in Linux

- WGET: retrieval of genome assemblies

```
wget http://ftp.sra.ebi.ac.uk/vol1/run/ERR333/ERR3335404/P7741_R1.fastq.gz
```

- CURL: retrieval of Bioinformatics files

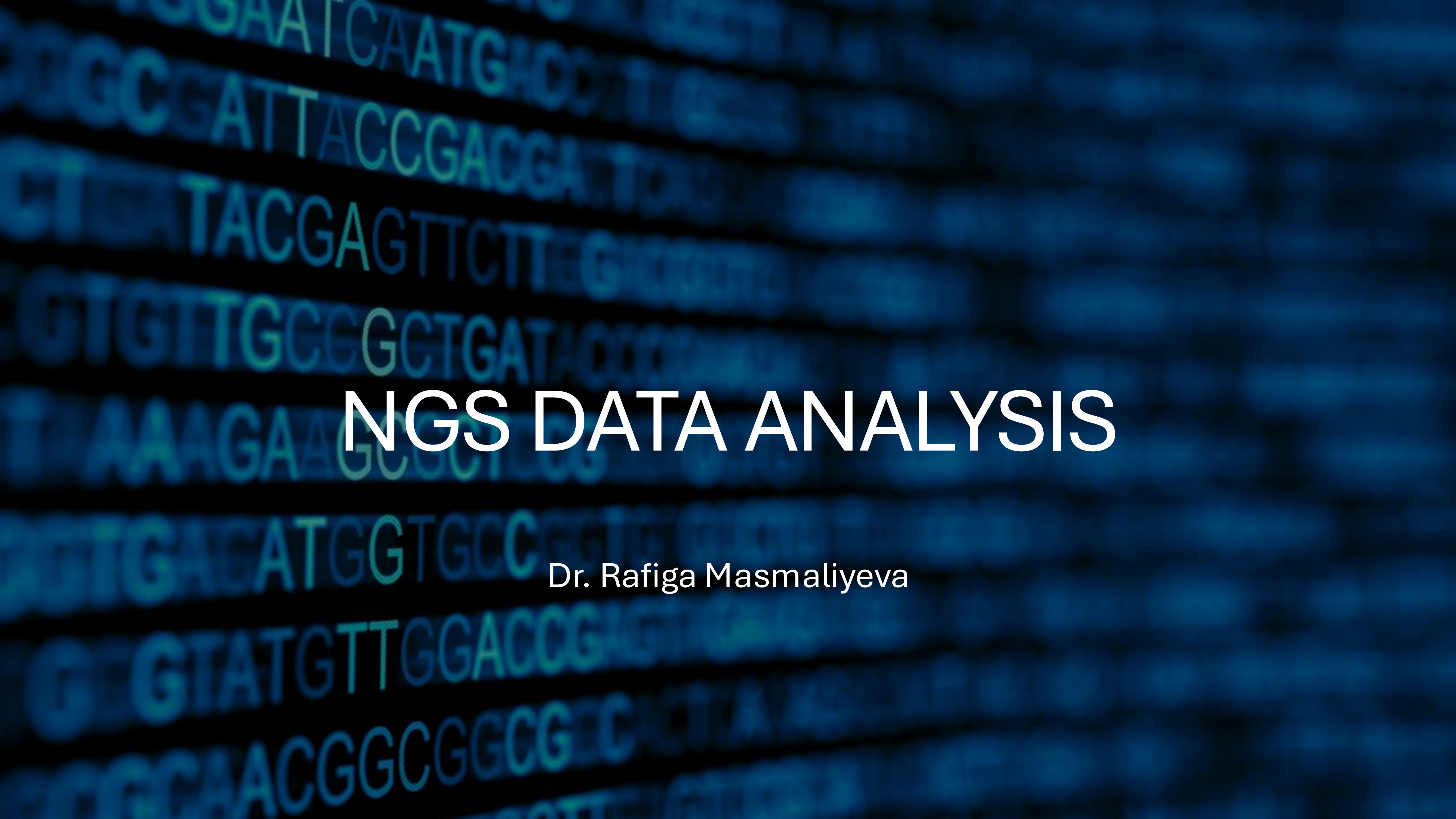
```
curl –O http://ftp.sra.ebi.ac.uk/vol1/run/ERR333/ERR3335404/P7741_R1.fastq.gz
```

- VIM: creation and editing of text files

```
head text.txt
```

- DIFF: comparing sequence differences in files

```
diff text1.txt text2.txt
```



NGS DATA ANALYSIS

Dr. Rafiga Masmaliyeva

NGS Data Analysis Workflow

1. Raw Data Quality Control (QC)
2. Trimming and Filtering
3. Alignment
4. Post-Alignment Processing
5. Variant Calling
6. Variant Filtering and Annotation

NGS Data Analysis Workflow

1. Raw Data Quality Control (QC)

- **Purpose**: Assess the quality of raw sequencing data to identify any issues early.
- **Tools**: FastQC, MultiQC
- **Explanation**: This step ensures that the data is of high quality before proceeding. It checks for issues like low-quality reads, adapter contamination, and GC content bias.

NGS Data Analysis Workflow

2. Trimming and Filtering

- **Purpose**: Remove low-quality bases and adapter sequences.
- **Tools**: Trimmomatic, Cutadapt
- **Explanation**: Trimming improves the overall quality of the data by removing poor-quality bases and adapter sequences that can interfere with downstream analysis.

NGS Data Analysis Workflow

3. Alignment

- **Purpose**: Map the reads to a reference genome.
- **Tools**: BWA, Bowtie2
- **Explanation**: Alignment is crucial for identifying where each read originates in the genome, which is essential for downstream analyses like variant calling.

NGS Data Analysis Workflow

4. Post-Alignment Processing

- **Purpose**: Refine the alignment to correct for errors and prepare for variant calling.
- **Tools**: SAMtools, Picard
- **Explanation**: This step includes sorting, marking duplicates, and indexing the aligned reads. It ensures that the data is in the correct format and free of artifacts that could affect variant calling.

NGS Data Analysis Workflow

5. Variant Calling

- **Purpose**: Identify genetic variants (SNPs, indels) from the aligned reads.
- **Tools**: GATK, Deepvariant, FreeBayes, Sentieon
- **Explanation**: Variant calling detects differences between the sequenced sample and the reference genome, which can be used for further analysis in research or clinical settings.

NGS Data Analysis Workflow

6. Variant Filtering and Annotation

- **Purpose**: Filter out false positives and annotate variants with functional information.
- **Tools**: VCFtools, bcftools, ANNOVAR, VEP
- **Explanation**: Filtering removes low-confidence variants, and annotation adds biological context, such as gene function and potential impact on protein function.

NGS Data Analysis Workflow

- https://cloud.wikis.utexas.edu/wiki/spaces/CoreNGSTools/overview
- https://mtbgenomicsworkshop.readthedocs.io/en/latest/material/day3/mappingstats.html
- Koboldt, D.C. Best practices for variant calling in clinical sequencing. Genome Med 12, 91 (2020). https://doi.org/10.1186/s13073-020-00791-w
- Austin-Tse, C.A., Jobanputra, V., Perry, D.L. *et al.* Best practices for the interpretation and reporting of clinical whole genome sequencing. *npj Genom. Med.* **7**, 27 (2022). https://doi.org/10.1038/s41525-022-00295-z