

Вежба 1 — Пробни рад: упознавање са Ubuntu/Linux CLI и GitHub Classroom (HTTPS + classic PAT)

Ова вежба служи да се упознате са окружењем у лабораторији, проверите да GitHub Classroom workflow ради како треба и да припремимо терен за следеће вежбе. Пратите кораке пажљиво — све је прилагођено томе да први пут радите у Ubuntu-у и терминалу.

Напомена: Предуслов је да сте завршили „Упутство 1“ (креиран GitHub налог, направљен **classic PAT**) и да сте прочитали „Упутство 2“ (HTTPS workflow у лабораторији).

0) Структура репозиторијума и очекивани излази

Репозиторијум за ову вежбу (**Vezba1**) садржи следеће фасцикле/фајлове (иницијално):

1. **bio_ml_script_test/**

Садржај: `bio_ml_env_selftest.py`

Задатак: покренути Python скрипту која прави JSON извештај и слике у `plots/`.

2. **bio_ml_notebooks/**

Садржај: `bio_ml_env_selftest.ipynb`

Задатак: покренути Notebook (Run All) који прави исти извештај/слике као и скрипта.

3. **awk_test/**

Садржај: `test_script.awk`, `dataset.csv`

Задатак: покренути AWK скрипту над CSV датотеком и уписати резултат у `awk_out.txt`.

4. **bioinfo_tests/**

Задатак: извршавати **линију по линију** команде из

`bash_script_test/bioinfo_test.sh`, ручно, и уписати резултате у одговарајуће `.txt` фајлове (в. испод).

5. **bash_script_test/**

Садржај: `bioinfo_test.sh`

Задатак: покренути ову bash скрипту да аутоматски направи исте `.txt` излазе као у

`bioinfo_tests/` (поента је да видите да низ команди може да се спакује у једну скрипту).

Очекивани излази (по фасциклама)

- **bio_ml_script_test/:**
`bio_ml_env_report.json` + подфасцикла `plots/` (PNG слике графика)
- **bio_ml_notebooks/:**
`bio_ml_env_report.json` + подфасцикла `plots/`
- **awk_test/:**
`awk_out.txt`
- **bioinfo_tests/:**
`blastn_test.txt`, `nhmmer_test.txt`, `clustalo_test.txt`, `seqkit_test.txt`,
`mafft_test.txt`, `samtools_test.txt`, `esearch_test.txt`
- **bash_script_test/** (након извршавања скрипте):
исти сет `.txt` фајлова као у `bioinfo_tests/`

1) Прихватање задатка и припрема радног директоријума

1. Отворите **URL позива** за задатак *Vezba1* у GitHub Classroom-у → кликните „Асепт assignment“ → сачекајте да се направи приватни репозиторијум у организацији **osnoviBioinformatike** (име ће бити `Vezba1-<ваш_github_username>`).
2. Отворите терминал и припремите локални директоријум (један студент по рачунару; за рад у пару видети одељак 6):

```
mkdir -p ~/Documents/lab-Vezba1-<vas_github_username>
cd ~/Documents/lab-Vezba1-<vas_github_username>
```

3. Клонирајте **HTTPS** URL свог репозиторијума и уђите у њега:

```
git clone https://github.com/osnoviBioinformatike/Vezba1-
<vas_github_username>.git
cd Vezba1-<vas_github_username>
```

4. Подесите **локални** Git идентитет (без `--global`):

```
git config user.name "<GitHub username>"
git config user.email "<ime.prezime@stud.bio.bg.ac.rs>"
```

2) Мини-увод у терминал (за потпуне почетнике)

Следеће команде користите кад год треба да се снађете у систему датотека:

```
pwd                # приказ тренутне путање
ls -la             # листај све фајлове/фасцикле (и скривене)
cd <путања>        # промени директоријум (нпр. cd bio_ml_script_test)
cd ..              # корак горе
cat <фајл>          # прикажи цео фајл у терминалу (за кратке фајлове)
less <фајл>         # преглед дужих фајлова (стрелице/space, Q за излаз)
head -n 10 <фајл>   # првих 10 редова
tail -n 10 <фајл>    # последњих 10 редова
mkdir <име>         # направи фасциклу
```

3) Задатак А — Провера Python окружења (скрипта)

1. Уђите у фасциклу `bio_ml_script_test/`:

```
cd bio_ml_script_test
ls -l
```

2. Покрените Python скрипту која прави извештај и графике у текућој фасцикли (JSON + plots/):

```
python3 bio_ml_env_selftest.py --outdir .
```

3. Проверите да су резултати направљени:

```
ls -l
jq .meta bio_ml_env_report.json    # кратак увид у метаподатке
ls -l plots
```

4. Вратите се у корен репозиторијума:

```
cd ..
```

4) Задатак В — Провера Python окружења (Jupyter Notebook)

1. Уђите у фасциклу `bio_ml_notebooks/` :

```
cd bio_ml_notebooks
```

2. Покрените Jupyter (ако је већ покренут, само отворите Notebook):

```
jupyter lab # или: jupyter notebook
```

3. У прегледачу отворите `bio_ml_env_selftest.ipynb`, изаберите kernel (нпр. „Python (bio_ml)“ ако постоји) и покрените **Run All**.

4. По завршетку, у терминалу проверите да је направљен `bio_ml_env_report.json` и фасцикла `plots/` у `bio_ml_notebooks/` :

```
ls -l  
jq .summary bio_ml_env_report.json
```

5. Затворите Jupyter и вратите се у корен репозиторијума:

```
cd ..
```

5) Задатак С — AWK тест

1. Уђите у `awk_test/` и погледајте улазни CSV:

```
cd awk_test  
head -n 5 dataset.csv
```

2. Покрените AWK скрипту над датасетом и упишите излаз у `awk_out.txt` :

```
awk -f test_script.awk dataset.csv > awk_out.txt
```

3. Проверите резултат:

```
wc -l awk_out.txt  
head -n 10 awk_out.txt
```

4. Вратите се у корен:

```
cd ..
```

6) Задатак D — Биоинформатички CLI тестови (ручне команде)

Циљ: да у `bioinfo_tests/` направите исте `.txt` излазе који се добијају аутоматски покретањем bash скрипте у следећем кораку.

1. Уђите у фасциклу:

```
cd bioinfo_tests
```

2. Отворите скрипту из „`bash_script_test/bioinfo_test.sh`“ да видите редослед команди (само преглед):

```
less ../bash_script_test/bioinfo_test.sh
```

3. Извршавајте **линију по линију** команде из те скрипте **ручно** (прескачите празне редове и коментаре који почињу са `#`).

- Ако у скрипти постоји преусмеравање излаза у фајл (нпр. `> blastn_test.txt`), користите га исто тако и овде.
- Ако неки ред позива алат без преусмеравања, сами додајте `>` `<одговарајући_фајл>.txt` да бисте добили тражене фајлове.

4. Када завршите, у фасцикли треба да постоје следећи фајлови (управо овде, у `bioinfo_tests/`):

```
blastn_test.txt
nhmmer_test.txt
clustalo_test.txt
seqkit_test.txt
mafft_test.txt
samtools_test.txt
esearch_test.txt
```

5. Вратите се у корен репозиторијума:

```
cd ..
```

7) Задатак Е — Биоинформатички CLI тестови (bash скрипта)

Сада ћете покренути унапред припремљену bash скрипту која треба да направи **идентичне** `.txt` фајлове, али у фасцикли `bash_script_test/`.

1. Уђите у фасциклу и дајте дозволу за извршавање (ако већ није извршно):

```
cd bash_script_test
chmod +x bioinfo_test.sh
```

2. Покрените скрипту:

```
./bioinfo_test.sh
```

3. Проверите да су генерисани `.txt` фајлови:

```
ls -1 *.txt
```

4. Поређење резултата из 6) и 7) (директоријуми `bioinfo_tests/` и `bash_script_test/`):

```
cd ..  
diff -q bioinfo_tests bash_script_test    # нема излаза = идентично
```

8) Предаја задатка (commit/push)

1. Проверите да је remote HTTPS:

```
git remote -v
```

Ако је SSH (`git@github.com:...`), подесите HTTPS:

```
git remote set-url origin https://github.com/osnoviBioinformatike/Vezba1-  
<vas_github_username>.git
```

2. Додајте све релевантне резултате и предајте:

```
git add bio_ml_script_test/ bio_ml_notebooks/ awk_test/ bioinfo_tests/  
bash_script_test/  
git commit -m "Vezba1: svi zadaci i izveštaji"  
git push  
# Username = vaš GitHub username; Password = vaš classic PAT
```

3. (Опционо) На крају сесије на лабораторијском рачунару:

```
git credential-cache exit
```

9) Рад у пару — брзи поступак (ако на једном рачунару раде два студента)

Ако радите у пару и немате времена да све кораке урадите два пута, можете радити **заједно у репоу студента А**, па потом пренети завршено стање у репо студента В (без копирања `.git/`).

1. Оба студента прихвате Classroom позив (да постоје оба репозиторијума).

2. Радите у `Vezba1-<githubA>` , завршите све кораке и урадите `git push` за студента А.
3. Пренесите садржај у `Vezba1-<githubB>` :

```
# са А → у радни директоријум В (без .git/ историје)
git -C Vezba1-<githubA> archive --format=tar HEAD | (cd Vezba1-<githubB> && tar
xf -)
```

или

```
rsync -a --delete --exclude ".git/" Vezba1-<githubA>/ Vezba1-<githubB>/
```

4. У репоу студента В направите commit/push са својим идентитетом (локални `git config user.name/user.email`).
5. На крају проверите да су резултати у оба репозиторијума комплетни.

10) Честа питања (FAQ)

- `git push` тражи „password“ → унесите **classic PAT**, не лозинку налога.
- Немате `jq` за преглед JSON-а → јавите асистенту у лабораторији (на библиотечким машинама је доступан).
- Jupyter не види kernel → ако радите на свом рачунару, пратите упутство за подешавање `bio_ml` окружења и `ipykernel` регистравање; у лабораторији је већ подешено.

Срећан рад!