

Course material for: GenEpi-BioTrain VT27: Clinical application of Nanopore-based 16S metagenomic sequencing for diagnostics

Project repo for following course on bioinformatics for 16S sequencing analysis on ONT, to be given at the ECDC course portal in June 2026:

[GenEpi-BioTrain - Virtual Training 27 - Clinical application of Nanopore-based 16S metagenomic sequencing for diagnostics](#)

Exercises

For the exercises, see the `/exercises` folder, or the links below:

1. [Exercise 1: Install and run Emu](#)
2. [Exercise 2: Troubleshoot Emu outputs](#)
3. [Exercise 3: Install and run Trana and TranaVy via install script](#)

Some GitHub codespaces caveats

If you are using GitHub codespaces to run the exercises, we noticed a few small caveats that can help improving the experience.

1. Sometimes the prompt string (the part just before your cursor), can get messed up so you don't see what you write. Ctrl+L doesn't even help then. What helps though is to clear the terminal with `clear`.
2. Sometimes the prompt can get so long that it is hard to read your own command. You can set a shorter prompt string, e.g. by running:

```
export PS1="[ \W]$ "
```

which will only print the current working directory in square brackets, and a dollar sign. An even simpler version with just a dollar sign, would be:

```
export PS1="$ "
```

Some caveats: Getting help

There are a number of ways to find documentation about commands in the terminal on Linux, of which some are not widely known. Below are some caveats to point you to what is available.

- `help` - The `help` command by default shows a very brief “cheatsheet” for commands that are not separate tools, but built-in functions in bash, such as for-loops, if-statements etc, which can be very useful, as their syntax is a bit different from other languages. Note also that you might want to pipe the output to `less` if it does not fit in your terminal: `help | less -S`. You can also specify the command you want help for, after `help`, e.g: `help for`, which will give slightly more verbose output.
- `man <command>` - Very many commands have a manual page, which you can read with the `man` command, followed by the command in question. Use your arrows to navigate, `/` to search (`n` for next and `N` for previous), and quit with `q`. Note that in the far end (which you can reach with `Home` or pressing `Page Down` a few times), there is often a section with examples, which can be very very useful.
- `<command> -h` OR `<command> --help` - The last resort, if you don’t find a help or man page for a command, is to try to use the `--help` or `-h` flag on the tool itself. This should hopefully give you at least some brief usage info and help.
- Pipe all to `less -S` - Sometimes the help screens from commands do for some mysterious reason get redirected to `stderr` instead of `stdout`, meaning that you can not pipe the content to a reader such as `less`, in order to be able to scroll and search it. This can generally be fixed by piping the content with `&` instead of just `|`. This will make sure that both `stdout` and `stderr` are piped via `stdout` to the next command, just as usual. Full example:

```
man -h |& less -S
```

More resources

- [Sandbox.bio](#) contains a number of great tutorials on common Linux commands and bioinformatics tools like `awk` and `seqkit`, where you can follow along and try out the commands in the browser. A few tutorials especially relevant to this course are listed below:
 - [Terminal basics](#)
 - [Wrangle FASTA and FASTQ with SeqKit](#)
 - [BAM parsing with samtools](#)
 - [Data exploration with awk](#)