Command Line Bioinformatics

Collected Notes

Part 1: Working on Linux

This part of the course is just about Linux: what it is, how it works, and how to use it productively.

Module 1.1: Getting Started

- What is Linux, Ubuntu?
- What is FOSS, GNU?
- Finding your way in the GUI
- Explore the software: LibreOffice, Firefox, ...
- Opening your first terminal, and what is `bash`?
- The file system and managing files: `cd`, `ls`, `mv`, `cp`, `ln`, `rm`
- Viewing and editing files: `cat`, `less`, `gedit`, `nano`, `vi`
- Just for perspective: `tac`, `nl`, `wc`, and much more ...
- Where to find help: `--help`, `man`, GIYF, AskUbuntu
- Shell conveniences: `history`, `!!`, `alias`

Module 1.2: Running Software

- Executables and the `PATH`
- Permissions and ownership: `chmod`, `chown`
- Globbing and filename patterns: `*` and `?`
- Escaping by Quoting: `"` and `'` (and `` ` ``)
- Redirection of input and output: `>`, `>>`, `tee`
- Pipes and filters: `|`, example: `cut`, `sort`
- Backgrounding: `^Z`, `bg`, `fg`

- Aborting: `^C`

Module 1.3: Linux Administration

- The User vs Administrator separation: `sudo`
- Updating: `sudo apt update && sudo apt upgrade`
- Installing software: the GUI, but mind the snap
- Installing software: `apt` (search, install, remove)
- Configuring software: in `/etc`
- And software that is not in Ubuntu: part 2

Module 1.4: Working Remotely

- Network terminology: the internet, addresses, ports, protocols
- SSH certificates
- Working remotely: `ssh`, `scp`, `rsync`, `ftp`
- Using `screen`
- Working on a cluster

Module 1.5: Shell Scripting

- A simple loop in the shell: `for .. in`
- Using variables: `HOME`, `PATH`, your own variables
- Your first shell (bash) program

Learn more?

- LinuxCommand.org
- Code Academy

Part 2: Bioinformatics Installation

In this part we learn about the various ways to install bioinformatics software, and in the process set up a bunch of useful tools.

Module 2.1: Installing Software

- The dependency management conundrum
- Containerisation to the rescue?
- User-level vs system-wide (`PATH` revisited)
- The importance of reproducibility

- Linux peace of mind: what to do if it breaks?

Module 2.2: "Classic" Installation

- Using `apt` always preferred: `samtools`, `bwa`, `bowtie2`
- Recent additions in Ubuntu: `spades` (but old), `unicycler`, `skesa`
- Installing a downloaded binary: ncbi-blast+ (`blastn`, `makeblastdb`)
- Installation as source: `unfasta`, `SPAdes` (source & run test)
- Installing from source tarballs: `make` (SKESA), `cmake` (MegaHit)

Module 2.3: Git and GitHub

- What is git, GitHub? (And GitLab, Bitbucket, SourceForge)
- Doing a `git clone`
- Example: install PROKKA
- Updating: `git pull`
- Reporting issues on GitHub
- Consider using `git` for your own benefit: `git init`, `git add`, `git commit`, `git log`.

Module 2.4: Docker

- What is Docker and when do I use it?
- Installing: `docker pull`, `docker build`
- Image vs container, managing: `docker image ...`, `docker container ...`
- Running: the `docker run -ti --rm -u $(id -u):$(id -g)` (what?!)
- Cleaning up: `docker {image,container} prune -f`

Module 2.5: Conda, Python, R

- What is Conda and when to use it?
- Peculiarities of Python and R: the need for venv, pip, etc
- Installing conda
- Installing in conda environment: `conda create ...` (Pangolin, pyani)
- Adding a wrapper script: `pyani`
- Conda channels full circle: back to dependency hell (and a pip tip, and mamba?)
- Miscellaneous topics: Snakemake (gplas)

Part 3: The CGE Tools / BAP

Module 3.1: FF lecture "Intro to Bioinformatics"

- Core concepts: alignment, assembly, mapping, k-mers
- How do the CGE Tools work
- Build your own AnythingFinder
- Practical: MyDbFinder, Abricate

Module 3.2: Preamble: doing the QC

- FastQC
- FastQScreen
- Illumina Report Viewer and Interop
- Trimming and contaminant cleaning (TrimGalore)
- MultiQC

Module 3.3: Setting up CGE Tools and BAP

- The CGE Tools
- The Databases
- The KCRI CGE BAP

Module 3.4: Running the Tools

- MLST
- ResFinder
- PlasmidFinder
- VirulenceFinder
- All-in-one: the BAP

Module 3.5: Experimentation time

- Nullarbor

Part 4: Programming your own

Module 4.1: Bash and friends get you a long way

- Bash programming: conditionals, loops, tests
- Regular expressions: `grep`, `sed`
- Swiss army knife: `awk`
- Practical: Build your own gene cutter

Module 4.2: Introduction to Python

- The Python REPL in the Terminal
- IPython3 in the Terminal
- Python in the Browser
- Writing a standalone program
- Numpy, SciPy, ...

Module 4.3: Using R and RStudio

- The R command shell
- Installing RStudio
- Using RStudio Server
- Using RStudio-based packages
- The Groningen AMR package?
- Writing standalone programs in R