Introduction

Don Teng

This first introductory page contains a bunch of essential readings to get the basics first. Useful sources: Wikipedia, Reddit Explain Like I'm 5 (ELI5), Khan Academy (for undergraduate-level academic concepts), Stackoverflow (a Q&A site for programming), Youtube.

Contents

This repo contains several tutorial files, with varying levels of difficulty. Recommended prerequisites will be listed near the top of that tutorial. However, try as I might to make them so, there's no guarantee that each tutorial is entirely self-contained, but the idea is that you should be able to start anywhere.

- `about-our-hpc` - How to use the lab's internal server, located in room 217
- `alignment1-introduction` - How to do multiple sequence alignment, part 1
- `alignment2-mafft` - How to do multiple sequence alignment (using `mafft`), part 2
- `m3-about-the-hpc` - WIP. Some resources, glossary of terms.
- `m3-quickstart` - A tutorial on how to send a job to M3.
- `m3-tips-and-tricks` - A dumping ground for how to use M3 slightly more efficiently
- `methods-tree-computation` - How to use various programs to compute a tree, and their pros and cons
- `misc-file-formats` - Tutorial on commonly used file formats like `fasta`, `phy`, etc.
- `misc-learning-to-code` - tutorial on how to *learn* to code, not actually on how to code. Has links for the latter.
- `misc-writing` - How to write so that others can understand you.
- `software-beast` - A link to existing BEAST tutorials.
- `software-paml-treesub` - How to install and use `PAML`, and an associated program, `treesub`.
- `software-python-resources` - A bunch of python resources.
- `software-raxml` - How to install RAXML, and basic usage.
- `software-tips-and-tricks` - A dumping ground for miscellaneous bash or computational tips and tricks.
- `tech-github` - Tutorial for basic command line usage of Github.

Things to Read up on

Mathematical Concepts

Recommended youtube channel: khan academy for undergrad-level theory, and mathematicalmonk for higher-level concepts. An in-depth knowledge of these concepts is not essential, unless you're aiming to specialize in that area - you certainly don't need to know how to do the maths by hand. An undergraduate understanding of these is sufficient; even Wikipedia is a little overkill.

- Linear regression
- Markov chains
- Maximum likelihood
- Bayesian statistics
- MCMC - the most difficult of the lot. Don't bother if you don't need to know this.

Phylogenetics Concepts

- The Wikipedia article on computational phylogenetics is a good starting point - it's sufficiently comprehensive that, at least, you'll be able to pinpoint what you don't know and look for that elsewhere. Also, admittedly, passive reading is a pretty dry and ineffective way to learn; there are "learn by doing"-style tutorials in the works.
- How to interpret a phylogenetic tree, by Andrew Rambaut. Or this Khan academy video.
- Models of DNA substitution, from Wikipedia.
- Hierarchical clustering. Otherwise known as "neighbour joining (NJ)" in phylogenetics literature. We don't use NJ trees very often, but it's a good conceptual starting point, and is easy enough to do by hand.

Programming

Having a good grasp on how to read, if not write, code is helpful, but not essential.

- Python, or R to start off. IMO, Python is superior to R in every way except for plotting. The Youtube channel thenewboston is a good place to start.
- Github. Here's a recommended video.
- Bash terminal, and Linux/Mac OS organization.

Software We Use

The following is a list of frequently-used-software. In general, try not to install computational software via ESS. The installation instructions shown here are only for Macs.

- **XCode and XCode Command Line Tools (for Macs)** - sets up your Mac for

development work. Get these first and foremost, because many of the computational programs require these to install properly.
- `homebrew` - A Mac OS package manager. Get it here. You'll need XCode first, and maybe XCode CLT as well. Homebrew installs many kinds of software, including scientific software, into the directories of your Mac automatically so you don't have to organize all your software. For instance, to install software, use:

```
brew install my_software
```

To see if `homebrew` has your software available:

```
brew search my_software
```

Many of the computational programs are executables, so using `homebrew` will also save you having to mess with your `$PATH` and all that (if you don't know what that is, all the more reason to use `homebrew`!)

- **MAFFT** - For multiple sequence alignment. An executable. Get it via `homebrew`.
- **RAxML** - For computing trees using maximum likelihood. This is available on M3, so installing this is optional. Allegedly available via `homebrew`, but I never managed to get the `homebrew`-downloaded version working. In any case, it's so terribly slow that you may prefer to send all RAxML jobs to M3 anyway. In any case, here's my RAxML installation and quickstart tutorial, because it's probably the clunkiest software on this list - difficult to install and difficult to use.
- **IQ-Tree** - For tree computation using maximum likelihood. An executable. Get it via `homebrew`.
- **FastTree** - Fast tree computation. An executable. Get it via `homebrew`.
- **BEAST package** - For tree computation using Bayesian statistics. Also has other secondary uses related to the understanding its own output, because the algorithms used to compute a Bayesian tree are not easy to understand. Version 1.8.4 is available here; that BEAST package also consists of accessory programs `Tracer` v1.6 and `BEAGLE` v2.1. Note that there's a version 2 available, though we're mostly still using version 1 so far.
- **FigTree** - a nice, lightweight program for tree drawing. Simple installation.
- **TempEst** - another nice program for tree drawing. We use it over FigTree because of one particular "find best root" function that's not available in `FigTree`.
- **CDhit** - For clustering DNA sequences by similarity. An executable; get it via `homebrew`.
- **PAML** - Multi-purpose analysis package with miscellaneous uses.
- **AliViewer** - Allows you to look at a `.fasta` file of sequence data.

Unfortunately, almost all the software we use is not very well-documented, or the documentation is written in a way that's only accessible to intermediate users. This creates an odd catch-22 situation where beginners are kind of stranded without even being able to RTFM. In which case, it may take a bit of experimentation to figure out the software behaviour; there will also be tutorials available in this repo to walk you through it. An exchange between my old supervisor at VLSCI and I:

```
me : Doesn't anyone teach you how to use Bash?
jni: Nobody teaches you how to use Bash; you're just expected to know it.
me : Wait, then how did the first person learn to use Bash?
jni: That first person was the guy who created bash, so he didn't have to learn it!
```

For Python users:

- Use Anaconda for automatic package management. Conda environments are also great for controlling your packages, and version control between Py36 and Py27.
- Useful packages: `pandas`, `numpy`, `Biopython`, `xlrd`, `scipy`, `scikit-bio`
- Jupyter recommended as an IDE.

Recommended Progression

Still thinking about this

1. Get used to the Terminal: learn about `homebrew`. Get `MAFFT` and `FastTree` first. These are good starting points because they're fast and (relatively) easy. Other programs like `RAxML` and `BEAST` are more involved.
2. Tree drawing with `FigTree`.

Other Recommended Reading

- *What Have You Tried?*, by Matt Gemmell.