PoreCamp2016 : Using the Linux command-line

0 Overview

By the end of this tutorial, you will have a basic understanding of computing concepts, be familiar with common computing jargon, and be able to navigate a Unix-like file system using command-line tools.

1 Text files

- All computer files contain a linear sequence of binary numbers, and are thus “binary” files.
- Text files are a special type of binary file where the numbers correspond to human-readable characters.
- The most common convention for interpreting particular numbers as characters is ASCII (the American Standard Code for Information Interchange).
- Another standard is Unicode, which includes numbers for characters for almost all characters used in all languages. Popular versions include UTF-8 and UTF-16.
- Text files do not contain the formatting information in other common files formats (e.g., Word).
- The end-of-line character in Unix/Linux/Mac is the Form Feed (\n), on Windows it is the carriage return followed by the form feed (\r\n) and on some older Macs it is just the carriage return (\r).
- Traditionally, text files are named with the suffix “.txt”, but this is not essential.
- Text files should be created using a (plain) text editor.

2 Text files in bioinformatics

- Used widely because they are human readable.
- Can be compressed further, if required (e.g., using tar+gzip, gzip, bgzip or zip).
- Character-delimited text stores data in a matrix, where each row contains the same number of “tokens” separated by a character (e.g., comma for CSV, tab for TSV, or any other character).
- File formats were devised so that you can write programs to deal with information that is stored in exactly the same way in each file to make “parsing” the file easy with programs or scripts.
- Common formats are: FASTA, FASTQ, GenBank, SAM/BAM, HDF5, and GFF.

3 Text editors

- Do not use a program like Word that will insert formatting characters around your text.
- Common editors include gedit, nano (easiest), vi, emacs, but there are many, many others.

4 The Unix/Linux/OSX command-line

- Unix is an operating system developed in the 1970s to work on “mainframe computers”, but can now run on anything.
- Linux is a version of Unix for “personal computers” (which contained different hardware components to “mainframe” computers) and is now the standard operating system for most servers, which these days, are usually clusters of PCs (not “mainframes”).
- OSX is a version of Unix.

- Unix, Linux and OSX are all very similar, but not identical. For our purposes, we can refer to all the commands we will be using as “Unix” commands.
- All commands are typed in a terminal window the context of an “environment”.
- There are graphical interfaces to the Unix operating system that are based on Windows, but originally, it was solely a command-line interface. Anything that can be done in the graphical interface can be done on the command-line.
- Since most bioinformaticians want to do similar operations on potentially thousands of files, with a sequence of commands stored in a file (i.e., a “script”), it is easier to do this at the command-line.
- All Unix commands were written by someone, usually in C, then “compiled” into a “program” that will run on the particular hardware and operating system of your computer.
- The Unix command-line has a “no news is good news” philosophy. If nothing gets printed after a command, it probably worked. If something doesn’t work, you get an error message.
- Being careful and understanding what you are doing before you hit “Return” is very important because Unix assumes the user knows what they are doing and will execute the command without asking you to confirm before doing so – so if you make a mistake, you might delete, destroy, overwrite or otherwise corrupt your data.
- There is no “undo” facility in Unix. If you remove a file, it is permanently deleted. If you edit a file, save your changes, and exit the editor, there is no way of retrieving the previous version of the file.

5 Programs and scripts

- Although a “program” can refer to any set of computer instructions written in any computer language, it now usually refers to a file of instructions written in a computer language that has been “compiled” into machine code (which is a sequence of 0s and 1s) that can then be “executed”.
- A “script” refers to a sequence of commands written in an “interpreted language” (like sh, bash, Python, Perl, R), rather than a “compiled language” (like C, C++ or Java).
- Almost anything you might want to do as a bioinformatician can be done with a combination of shell, Python (or Perl) and R scripts. The only reason to do it in a compiled language like C++ is for the code to run faster processing large numbers of massive files more quickly (because opening and closing files, and iterating through files is faster in a compiled program).

6 The Unix file system is hierarchical

- The Unix file system consists of directories and files, some metadata (e.g., name, date+time of last access, size in bytes) is stored with each.
- The directory separator on Unix systems is the forward slash “/”.
- All directories and files are accessible via “absolute” (that start with “/”) or “relative” path names.

7 Common Linux commands

Command	**Description**	**Examples**
cd	Change directory	cd

		cd $HOME/work/presentations cd 2015/10/logs cd ../conferences
ls	List directories and files	ls –l ls –l work/presentations/2015/*.docx ls –las ls –lrt /PATH/TO/DATA ls –lR work/presentations
pwd	Print working directory	pwd
grep	Search by string pattern	grep "^>"
open	Infer the file type and open it using an appropriate program (OSX only)	open figure1.pdf open agenda.xlsx open olddata.zip
awk	A text-processing language	ls -l | awk '{print $1}'
find	Find files that match a pattern	find . –name todo.txt –print 2>/dev/null find $HOME/work -name "*.fasta" –print find /PATH/TO/DATA –name “*.fasta” –print
man	Display the manual page	man grep man man
ssh	Open a secure connection to another computer	ssh SERVERNAME ssh USERID@SERVERNAME
scp	Secure copy	scp todo.txt SERVERNAME: scp todo.txt SERVERNAME:/home/USERID/Desktop scp USERID@SERVERNAME:/home/USERID/dataids.txt .
locate	Find filenames quickly	locate samtools
ftp	Copy a file to another computer	
which	Locate a program file in the user's path	which bash
perl	A programming language	perl –p –i –e “s/^>gi/>G1_gi/g” sample.fasta

8 More information

A Google search will turn up many tutorials for learning Linux/UNIX commands, for example:

- http://www.ee.surrey.ac.uk/Teaching/Unix/
- http://manuals.bioinformatics.ucr.edu/home/linux-basics

There are also many books about computing and bioinformatics for biologists, for example:

- Practical computing for biologists (Haddock & Dunn, 2011)