Texas A&M University

Department of Electrical and Computer Engineering
Dwight Look College of Engineering

BIOL 689 Digital Biology

Zhiyang Ong [1]

Notes from the class and revision notes

June 23, 2014

[1]Email correspondence to: ongz@acm.org

Abstract

Notes from my BIOL 689 Digital Biology class at Texas A&M University in the first summer session of 2014.

To be completed...

Revision History

Revision History:

1. Version 0.1, June 3, 2014. Initial copy of the report.
2. Version 0.2, June 23, 2014. Added material for the first lecture.
3. Version 0.3, June ???, 2014. Added BLAH and modified BLAH:
   (a)
4.

Contents

Chapter 1

Introuctory Material

This is a UNIX-based class.

My username is db00XX. See the comments of this statement for my username.

Ricardo is my (lab) teaching assistant (TA).

Genome assembly is still an unsolved problem.

GitLab (from GitHub, Inc.) will be used for the first time in this class. Three concepts will be covered in the introduction: Wiki for the class, which contains the standard class information; the code repository that the Wiki uses; and the *Git* version control system.

The outline of the class (i.e., syllabus and class schedule) will be modified as the semester progresses.

On Thursday, June 5, 2014, we will cover genome analysis, gene models, and gene files. Next week, we will cover next generation DNA sequencing. We will also look at library construction methodology and techniques, and associated challenges. Next Thursday, we will also look at "small reads." We will write small scripts to process small data sets, and organize the pipeline (or design the algorithm) for the program/script. That is, design the control and data- flow graph of the algorithm. Furthermore, we will look at genome mapping, genome assembly, data display (i.e., data visualization), and transcriptome mapping and transcriptome assembly. Subsequently, we will be given data sets from the professor to carry out (machine) learning for pattern classification, and explore read archives (with unknown outputs).

Chapter 2

Software Engineering Basics

2.1 UNIX Basics

My TA, Ricardo, suggested using Guake (http://en.wikipedia.org/wiki/Guake) as a substitute for the common/normal `Terminal` application.

We will be using the `Terminal` to do a lot of our work in this class. Prof. Rodolfo Aramayo briefly talked about the history of UNIX and its derivatives, such as *Linux*, *BSD*, *Oracle/SUN Solaris*, and *Mac OS X*. UNIX was started at *Bell Labs*. He also talked about the UNIX philosophy.

We shall operate in the UNIX environment via text files. Everything (including directories) is a file in UNIX. Some files can be read visually (i.e., text files), while others (i.e., binary files) cannot.

The kernel is the heart of the operating system. The UNIX shell (accessed via applications, such as the `Terminal`) is an application that allows users to interact with the kernel indirectly.

Anatomy of UNIX commands: `command_name [options] [arguments]`. Double dashes for options of UNIX commands cannot be combined. However, for options for single dash lines, they can be combined.

The "`man`" page is the UNIX manual. To find documentation of a UNIX command, use the `man` command.

`SSH` is an application that allows me to connect securely to another computer that is connected to the same computer network, or to the Internet.

`rsync` is an application for file copying and synchronization between different computer accounts. It does not copy all files in your directory, but copy modifications to existing files and copies only new files. It transfers files in compressed format.

UNIX commands to learn:

1. alias: "alias ll"
2. apropos: "apropos copy" would search the UNIX "man" pages for the keyword "copy".
3. cat: conCATenate
4. cd

5. chmod: Change mode
6. clear
7. cp: cp –version
8. dir -l
9. date
10. du: "du -hd 0 ." list the size of the directory in KB, and "du -hd 1 ." list the size of the directory and its files. "df -h" indicates the size of the directory and its contents.
11. echo: "echo -e" refers to `echo` enhanced, which redirects the output in the UNIX pipeline to a file. *echo -e " 'date'" > tata1.* "echo $PATH"
12. file
13. history
14. info cp
15. less
16. ls [-al]
17. more
18. mkdir
19. mv
20. pwd
21. rm
22. rmdir
23. rsync. An example of how the command can be used is: "rsync -v username@host:∼/path/to/file .". The "-v" option runs the UNIX command in verbose mode.
24. touch
25. tree
26. type: type zrio
27. whatis
28. which: which blastn

Use "tab" to autocomplete filenames and directory names. Avoid using spaces in filenames and directories to keep file and directory access simple.

Directory access: The "." file is the current working directory, and the ".." is the parent directory. A directory can also be called a folder. By using the `cd` command, I can return to my home directory.

You cannot undo operations in UNIX. Hence, save and backup files before performing removal operations in UNIX. There is also no "trash can" or "recycle bin".

Microsoft Excel has a maximum limit of 65,000 rows in the spreadsheet. Hence, this limits the amount of information that I can process with Microsoft Excel. To process more data, such as GBs or TBs of data, I need other software applications or develop my own computer program.

Symbolic links in UNIX are like shortcuts or aliases in Windows. An example of creating a symbolic link is: "ln -s ../01/test01".

The human genome has been decoded into a file about 7 TB.

The colon ":" serves as a dummy placeholder to remove the contents of a file; ": > filename"

Standard output stream, `stdout`, is described along with exit signals of UNIX processes. Standard error output stream will write to the standard error output file. UNIX redirection for standard output and error streams are described.

Use *tree* to show contents of a directory as a tree.

Discussed UNIX path redirection, pipelining of UNIX commands, and separate execution of UNIX commands (using the semicolon ";" symbol).

Covered special/escape characters to use tabs and newlines to print information.

Covered information on how to go to the "home" directory. "∼" refers to the home directory.

Covered absolute paths and relative paths in UNIX.

Detailed explanation of the "ls" command. It indicates when the file has been created/modified. It also indicates the size of the file in bytes. It also indicates the username ("db0015") and the group ("student") that I belong to. Permissions to access files are determined by the group that I belong to. File permissions are indicated for read, write, and execute. They are set for individual users, groups, and everybody with access to the computer network/system. File types are indicated for directories ("d"), regular/normal files ("-"), and symbolic links ("l").

Most files have the file permissions set as 755.

Discussed how to create aliases in UNIX.

Configure my UNIX environment with the ".bashrc" (or ".bash_profile") file.

GUI-based *Galaxy* is used for this class.

2.1.1 SSH Basics

File

There are many applications for downloading files from the Internet. The applications `curl` and *wget* are more common for downloading files.

The UNIX command `ifconfig` gives you information about computer networking for your computer or computing account (if you are connected to a remote computer).

There are many

2.1.2 Shell Scripting Basics

Download data to group directory on "Geiger", so that I do not corrupt the local machine.

Use "tree" to find out the directory structure of the specified directory.

For class on June 17, 2014, clone the repository from Prof. Aramayo, https://geiger.tamu.edu/gitlab/raramayo/digitalbiology_project_summer2014. Work on this directory to practise the UNIX sub-lesson for today.