BST281: Genomic Data Manipulation, Spring 2017

Monday 06: File utilities and command environments

This lecture focuses on performing basic data manipulation and analysis at the command line. Many simple questions can be answered by chaining \*nix programs (without the need for a custom program). File manipulation at the command line is faster than a graphical interface, but less forgiving of mistakes (e.g. accidental deletion). Back up your work!

**Command line basics**

* Many small programs highly optimized for a particular task.
* Chain programs (via piping) to accomplish larger, more complicated tasks.
* These programs are available by default in the Linux/MacOS terminals, and can be added to Windows with (e.g.) CygWin or GOW (Gnu on Windows).

**Command line programs**

* Discovery tools
  + man *program* – displays information about how to use a tool, including flags. Q to quit.
  + ls – list the files in a directory, or matching a particular wildcard pattern (e.g. \*.txt)
* File manipulation
  + rm *path* – delete a file (path) permanently.
  + cp *path1* *path2 –* copy a file from one location to another (overwriting without warning).
  + mv *path1* *path2 –* move a file from one location to another (also used to rename a file).
* View data
  + cat – stream the content of one or more files to the screen (STDOUT).
  + less – view the content of a data stream, often at the end of a chain of commands. Q to quit.
  + diff *path1* *path2 –* view differing lines between two files, or confirm that they are identical.
* Subset data
  + head/tail – view the first/last 10 lines of a file/data stream.
  + cut -f*N* – slice tab-delimited columns from a data stream (here, column *N*). 1-based counting.
  + grep *pattern –* subset rows of a data stream that contain (match) *pattern*.
* Reorganize data
  + sort – sort the lines of a data stream (lexically by default, use “-n” for numeric sort).
  + sed “s/*find*/*replace*/g” – perform find and replace operations within lines.
* Summarize data
  + wc – count the lines, words, and characters in a datastream (wc –l for lines only).
  + uniq – isolate unique, adjacent lines of a file (use sort | uniq to force global uniqueness).

**Building a chain of commands**

* Start the chain by 1) cat’ing one or more files, 2) providing a file (path) as an argument to a program, or 3) feeding a file to a program as STDIN with “<” (as in *program* < *path*).
* Use the pipe (“|”) to feed the STDOUT of one program as the STDIN of the next program.
* End the chain by 1) viewing the last output in less or 2) redirecting to a file with “>”.

**Python: the subprocess module**

* Import subprocess to invoke command-line tools within Python and capture their output.

**Suggested reading**

* Haddock and Dunn, Chapter 4 p47-66, Chapter 5 p67-81, Chapter 16 p299-309
* <http://docs.python.org/library/subprocess.html>
* [http://linuxcommand.org](http://linuxcommand.org/)