Programming for Bioinformatics - part 3

Saul Pierotti

January 15, 2020

UNIX

- Unix was developped in 1960 at Bell labs by the founders of C
- It was one of the first OS to be multi-tasking, multi-user
- It has a hierarchical file system
- In the root directory we can find
  - `/bin` contains essential user command binaries
  - `/etc` configuration files
  - `/sbin` contains essential system binaries
  - `/usr` contains binaries and support files for user apps
  - `/var` contains variable data files
- It is written in C
- This part is about shell scripting
- Unix commands are mostly similar everywhere, but sometimes there are differences
- Alisases can be used for typing frequently used command parameters
  - They can be removed with the unalias command
  - They can be made permanent by putting in the .bashrc

Shell

- The shell is a language interpreter
- When I type a command, it searches for the command in what is in the `$PATH` variable
  - `/bin  /usr/bin  /usr/local/bin`
- In order to execute commands that are not in `$PATH`, i need to give the path
  - `./myscript`
- I can write on multiple line by putting before pressing enter

File permissions

- They work for any file (also directories, which are indeed files)
- The fundamental permissions are r, w and x and they can be applied to owner, group and all
- The combination of permission of a file are represented with 3 bits for a single user
  - 000 is no permission, 100 is r–, 010 is -w-, 001 is –x and so on
- I can express a permission status by specifying 3 numbers, and so using octal numbers
  - 0 in octal means 000 in bynary, so it is —
  - 1 means 001, so –x
  - 7 means 111. so rwx
  - 000 is ———- or d———
  - 777 is -rwxrwxrwx or drwxrwxrwx
  - 345 is –wxr–r-x

Some commands

- `echo` is the Bash way for print
- Print the working directory: `pwd`
- Create a directory: `mkdir`
- Create an empty file: `touch`
  - If I touch an existing file, I change its acces and modification time
- Copy files or directories: `cp`
  - For doing recursively (for dirs) use `cp -r`
  - It can overwrite: use `cp -i` to ask for confirmation!
    - I can also make an alias `cp=cp -i`
- Remove files: `rm`
  - There is no confirmation!
  - `rm -r` is recursive
  - `rm -i` asks for confirmation
- Remove empty directories: `rmdir`
- Move or rename: `mv`
- Scroll a file:`less`
  - I can search for words in less with `\something`
  - I can exit with `q`
  - `more` is a primitive version of `less`
- Search a file: `find`
  - I write first the directory in which I want to search and then, for instance, the name of the file
  - `find . -name myfile.txt`
  - I can also search by size, permission (`-perm`)
- Display the manual: `man`
- Path of a command: `which`
- All the paths to a command and associated files: `whereis`
- Quick one-line info on a command: `whatis`
- Info on a file: `file`
  - It tryes to guess the filetype based on its content
- Free disk space: `df`
- DIsk usage stats: `du`
- For both `df` and `du` the `-h` option makes the output human-readable
- Reverse a string: `rev`
- Simple calculations: `bc`
  - In order to operate on reals instead of integers, I should use `bc -l`

File compression

- There are many tools and hence formats
- `gzip` and `gunzip` are used for .gz files
- `tar cfz` and `tar xfz` are used for .tar
- `zip` and `unzip` are used for .zip

Network utilities

- Connect to a remote machine: `ssh`
- Copy remote files : `scp`
  - It is called secure copy
  - `scp user@remotelocation.org:path/to/file /destination/path`
- Download from the web: `wget`
  - It works with http and ftp urls

Globbing

- The Unix shell provides wildcards that can be used to specify filename patterns
  - `*` matches any number of characters, also none
    - `echo *` is equivalent to `ls`
  - `?` matches a single character
  - `[abc]` matches a, b or c
  - `[!abc]` matches not (a, b or c)
  - `[a-z]` matches any single letter
  - There are some special patterns like `[:lower:]` or `[:digit:]`
- I can specify more than 1 pattern in the same line
  - `A* T*` is equivalent to `[AT]*`
- Brace patterns can also match non-existing filenames
  - `{A,B,C}{A,B,C}` is expanded to all the 2 characters combinations of the 2 lists
  - It would be `AA AB AC BA BB BC CA CB CC`

Redirection

- In Unix devices (printers, screen output, ecc.) are treated as files
  - The `stdout` and `stderr` devices are connected to the monitor
  - `stdin` is connected to the keyboard
- `stdout` is redirected with `>`
- `stderr` is redirected with `2>`
- I can append instead of overwrite with `>>` or `2>>`
- I can redirect all the output with `&>`
  - Be careful, `&>>` does not work on all systems (!)
- The standard way to append all the output is to redirect `stderr` to `stdout` and then append it
  - I can use `ls >> file.txt 2>&1`
- I can trash an output by redirecting to `/dev/null`
- `stdin` can be redirected with `<`
  - It is almost useless, and it can not work with some commands
- Pipe (`|`) is used for redirecting the `stdout` of a command to the `stdin` of another
  - It is used for building pipelines (!)
- If I want to store an intermediate result in a pipeline, I use `tee`
  - `input command1 | tee output1.txt | command2 > output2.txt`

Text manipulation

- Concatenate and print to stdout: `cat`
  - `cat file1 file2 > file3` creates file3 containing the concatenation of file1 and file2
- Print the first/last n lines: `head -n` and `tail -n`
  - `head -4 myfile.txt` prints the first 4 lines
  - I can print a specific line by piping head and tail
    - `head -4 myfile | tail -1`
- Sort a file content: `sort`
  - The default sorting behaviour is lexicographic: 10 comes before 1
  - If I specify to sort according to a specific column I specify -k
  - columns are defined by whitespaces
  - `sort -k 2 myfile` sorts according to the second column
  - I can sort numerically with `sort -n`
  - I can remove duplicated lines with `sort -u`
    - This works only if the lines are next to each other ofter sorting (!)
- Report or omit repeated lines: `uniq`

  - It detects only adiacent duplicates (!)
    * It is algortimically complex to detect unsorted duplicates
  - `uniq -d` prints only duplicated lines
- Extract a column from a file: `cut -f`
  - `-f` specifies the field separator, that defaults to Tab
- Count the newlines: `wc -l`