Unix cheat sheet

Alexander Jueterbock[*]

May 14, 2015

Being familiar with the Unix commandline is essential for bioinformatics data analysis but it might seem complex or tedious at the beginning. However, you will get already quite far with only a few commands. Here, I provide a cheat sheet that gives a quick overview of the most essential commands which will increase your efficiency in data analysis and file handling drastically - promised! You will find more details on most of these commands in my previous blog posts that were marked with the 'Commandline' tag.

1 FILE system

`cd DIR`	change directory to DIR
`cd ..`	go up one directory
`cd ~`	to to your home directory
`pwd`	show present working directory
`ls`	list items in current directory
`ls -a`	list all items, including hidden ones
`ls -lhcrt`	list all items in long, human-readable format and sort in reverse order by modification time
`ls -F`	list all items in current directory and show directories with a slash and executables with a star
`tree -C`	print hierarchical structure of your FILEs and directories (color-coded)
`tree -d`	print hierarchical structure of all subdirectories
`tree -sh`	print hierarchical structure of FILEs and directories with sizes (-s) in a human-readable format (-h)
`mkdir directoryname`	make new directory named directoryname
`mv FILE1 FILE2`	Rename FILE1 to FILE2
`mv FILE1 ../FILE2`	Move FILE1 one directory up
`cp FILE1 FILE2`	copy FILE1 and save it as FILE2
`rm FILE`	remove FILE
`rm -r DIRECTORY`	remove directory and all of its contents

[*] (Alexander-Jueterbock@web.de)

2 Opening FILEs and extracting information

less FILE	open FILE and scroll through it line by line
wc -l -w -m FILE	Counting lines, words, and characters in FILE
grep "pattern" FILE	Print lines from FILE that contain "pattern"
grp -v "pattern" FILE	Print lines from FILE that do not contain "pattern"
cat FILE > FILE2	Write the content of FILE to FILE2
cat FILE ≫ FILE2	Append the content of FILE to FILE2
sed -n 11,12p FILE	Extract lines 11 to 12 from FILE
awk -F "\t" ’$1 > 20 {print $0}’ FILE	Print all columns of a line ($0) in FILE if the value in column 1 ($1) is bigger than 20
unzip FILE.zip	unzip the zip-compressed FILE
gunzip FILE.gz	unzip the gz-compressed FILE
sort -n NUMBERS	sort a row of NUMBERS numerically
uniq -c FILE	count unique lines in FILE
nano FILE	open FILE on the command-line
xdg-open FILE	open FILE with the standard program for its file type
eog FILE	open FILE (which is a figure) with the Eye of Gnome graphics viewer program

3 Data transfer

rsync --progress -avz SRC DEST	transfer from SRC to DEST, show the progress while FILEs are compressed during the transfer in archive mode (including recursing into directories)
rsync FILE user@host://home/usr/	transfer FILE to the folder /home/usr on the remote server user@host
rsync -avz directory/ DEST	transfer all FILEs saved in directory to DEST
rsync -avz directory DEST	create the folder directory in DEST and transfer all FILEs in this directory
scp -r SRC DEST	transfer all FILEs in SRC to DEST
scp FILE DEST	transfer FILE to DEST

4 Executing scripts and programs

nohup ... &	execute ... in the background
nohup ... > FILE.txt &	execute ... in the background and redirect output to FILE.txt
ps -p ID	print the status of a process with the specified process-ID
kill ID	stop the process witht the specified process-ID
pkill NAME	stop all processes with NAME (NAME could be for example ’R’ or ’python’)
top	provides an ongoing look at processor activity in real time

5 Networking

`ssh user@host`	connect to host as user
`ssh -X user@host`	connect to host as user with X11 forwarding enabled (you can open programs with graphical user interface)

6 Help

`command --help`	Lists the options for command
`man command`	opens the manual page for command (exit with 'q')

7 Tricks

Pipe output from one command with `|` as input to another command.

`TAB key`	auto-completion of commands, FILE names etc.
`UP or DOWN arrows`	move through the history of your commands
`history`	Get overview of the commands you have used
`*`	Allows to generalize file names. For example, *fasta refers to all fasta files in a directory

Emacs 24.3.1 (Org mode 8.3beta)