Bash cheat sheet

Bulk RNAseq course

March 2019

Navigation

Command	Description
`man`	Usage manual of any command. Example: `man cd`will description, parameters and usage of the `cd` command
`pwd`	Print working directory
`cd`	Change directory (expects path)
`cd ~`	Change directory to `/home/user/`
`cd /`	Change directory to `/home/`
`ls`	List content
`mv`	Move file or folder (expects input and output)

Creation/Deletion

Command	Description
`mkdir`	Create directory*
`rmdir`	Remove empty directory*
`rm`	Remove file* (can use recursively for directories with `-r`. Careful!)
`touch`	Create empty file*
`nano`	Terminal file editor (One of many editors. `Ctrl-x` to quit)

*= Expects path/name

File manipulation

Command	Description
`cat`	Concatenate files. Prints content into terminal*
`cp`	Copy (expects a path/file and the final location)
`less`	Visualize contents of a file* (hit `q` to quit)
`head`	Print first 10 lines of a file*
`tail`	Print last 10 lines of a file*
`grep`	Print lines with specific motif (usage: grep motif file)
`wc`	Word counter (use the `-l` flag to count lines instead, or `-c` to count characters)*
`sort`	Sort alphabetically (use the `-n` flag to sort numerically)*
`uniq`	Print only unique lines (tip: sort them first)*
`chmod`	Change mode. Edit read, write and execution permissions of a file. Example: `chmod +x file.sh` to make a file executable*

String manipulation

Command	Description
`echo`	Print back whatever you input it.
`tr`	Translate. Example: `tr U T` will change all Us to Ts), or use the `-d` flag to delete characters. Example: `tr -d ";"` will remove all semicolons.
`sed`	Stream editor. Its basic usage replaces a string with another. Example: `sed 's/U/T/g'` will change all Us to Ts.
`cut`	Cut out sections of a file. Use the `-f` flag to get a specific field, and `-d` to specify the delimiter. Example: `cut -d \t -f 1` will get the first column from a tab delimited file

*= Expects path/name

Loops

Command	Description

`for`	Runs while the variable satisfies the condition. A couple of examples:
	
	`for i in 1..5; do`
	` echo Hello world $i`
	`done;`
	
	The loop will assign the variable "i" the value of one and will increase it till five, one per iteration. This loop will print "Hello world" five times with its iteration number.
	
	`for i in file.txt; do`
	` echo $i`
	`done;`
	
	The loop will assign to the variable "i" the value of each of the lines in the file and print it, till there are no more lines.

`while`	Another type of loop. It also runs while the condition is met.
	
	`while true; do`
	` echo Hello world`
	`done`
	
	Will print "Hello world" forever (the condition is always "true").
	
	`cat file.txt | while read line; do`
	` echo $line`
	`done`
	
	Will print each line of the file.

*= Expects path/name

Key single characters

Command	Description
`#`	Comment. Everything in the same line after this symbol will be ignored.
`|`	Pipe. It passes the output of a command as an input for another one. Example: `head file.txt | grep A` will print the lines containing "A" from the first ten lines of the file.
`>`	Take the output and print it into a file. Example: `head file.txt > head_file.txt` will print the first ten lines of the file into `head_file.txt`. If `head_file.txt` already exists, it will be overwritten. To print the result at the end of a file use `head file.txt >> head_file.txt`
`*`	Wildcard. It is replaced with anything. Example: `ls *.txt` will print all text files in the current working directory
`$`	Pointer to variables (like in the loops). Example: if we run `i=10` and then `echo $i` it will print `10`
`\`	Scape character. Special characters (underscore, slashes, and dollar signs before variable names) need to be after a backslash for them to be taken as strings. Example: `echo $i` would print the value inside i, but `echo $i` will print "$i"

Some slurm commands

Command	Description
`module spider`	Print description of the software. Example: `module spider bedtools`
`module load`	Loads the specified software into the session. Example: `module load bedtools`
`squeue`	List the jobs in the server's queue
`sacct`	List the jobs that I have submitted in the day and how they resulted (completed/failed)

*= Expects path/name