

BioUnix CheatSheet

Type	Cmd	Usage	Description
Nav	man	man cmd	Show help on a command
	pwd	pwd	Show current path
	ls	ls	List things in directory
	cd	cd dirname	Move to folder named dirname, use <code>cd ..</code> for previous dir
	mkdir	mkdir dirname	Make a new directory
Files	head	head filename	Print first 10 lines of a file, use <code>head -n 20</code> for 20 lines
	tail	tail filename	Print last 10 lines of a file, use <code>tail -n 20</code> for 20 lines
	cat	cat f1 f2 > f3	Merge vertically f1 and f2 into f3
	cut	cut -f1,2,5 file1	Print columns 1,2 and 5 of file file1
	paste	paste f1 f2 > f3	Merge horizontally file f1 and f2 into f3
	wc -l	wc -l f1	Count number of lines in f1
	sort	sort f1	Sort a file according to first column
	less	less f1	Open huge file f1 for reading, press \uparrow , \downarrow to move, q to quit
	grep	grep pattern f1	Print line where pattern found in f1
Cmd	nano	nano file.txt	Edit a file from the terminal (see later for how to)
	 	cmd1 cmd2	Dump (pipe) output of cmd1 to cmd2
	>	cmd0 > f1	Dump results of cmd1 to file f1

Some practical examples

Count element in a fasta file

```
grep -c ">" file.fasta
```

Count lines in all files ending with .bed in current dir

```
wc -l *.bed
```

Sort a bedfile by chromosome pos

```
sort -k1,1 -k2,2n bedfile.bed > new_bedfile_sorted.bed
sort -k1,1 -k2,2n -o bedfile.bed bedfile.bed
```

Remove extra columns in bedfiles (keep 1,2,3)

```
cut -f1,2,3 bedfile.bed > new_bedfile.bed
```

Remove lines with a specific pattern

```
grep -v "pattern" file > new_file
```

Keep lines with either of patterns p

```
grep "p1\\|p2\\|p3" file > new_file
```

Keep lines with patterns p1 and p2

```
grep "p1" file | grep "p2" > new_file
```

General tips

- Use the manual when you don't know. To do so: `man command`
- Use `<tab>` to autocomplete a command or file name
- Use `*` to match everything - i.e. `ls *WT*` show all files with WT in it
- Pipe results of command in head to check the results before running the full command: `grep ">" f.fa | head`

Bedtools intersect

Bedtools has a beautiful and well documented web page at <http://bedtools.readthedocs.org/latest/content/bedtools-suite.html>. It can do *everything* related to bed files. Here I only speak about intersect as it is the most used one.

Report if elements in bed1 overlap bed2 with 1 bp

```
bedtools intersect -u -a bed1.bed -b bed2.bed > bed1_overlap_bed2.bed
```

Report if elements in bed1 NOT overlap bed2 with 1 bp

```
bedtools intersect -v -a bed1.bed -b bed2.bed > bed1_N0overlap_bed2.bed
```

Report if elements in bed1 overlap bed2 with 50% overlap

```
bedtools intersect -u -f 0.5 -a bed1.bed -b bed2.bed > bed1_overlap50_bed2.bed
```

Intersect HUGE bed files

```
#Sort the two bedfiles first using sort as stated before  
bedtools intersect -sorted -u -a bed1.bed -b bed2.bed > bed1_overlap_bed2.bed
```

awk filtering

Awk is a program to process large file. Basically, it will run through the file and execute a command on each line. It is possible to add, swap or remove columns, do computation across lines and so on.

Keep if columns 5 is 'LTR'

```
awk '$5 == "LTR"' file.txt > file_ltr_only.txt
```

Keep if columns 2 is bigger than 42

```
awk '$2 > 42' file.txt > file_42.txt
```

Use tab as a separator

```
awk -v OFS='\t' 'print $1,$2,$4,$5,$3' > file_w_swapped_columns.txt
```

Text edition in a terminal

To edit a file on the terminal you have to use a terminal based text editor. Most known ones include Vim, Emacs and Nano. As the latter is *by far* the easiest, you should use nano.

Edit a file

```
nano file.txt
```

Move around and edit

If the file do not exist, it will be created.

Move around and edit

In some terminal, the mouse do not work. Use the arrows to move around the document and edit what you need. Type text as in any regular editor.

Save and exit

At the bottom of nano, the shortcut are displayed. The `^` is used to symbolise **Ctrl**. Use `ctrl-o` to write the file. `ctrl-x` to quit. Type `ctrl-g` for more help.

SSH

SSH is a protocol used to connect remotely to another computer. When working on vital-it, you will do it through ssh.

Connect to a server

```
ssh user@server.com
```

Connect using X11 (allows to display some things)

```
ssh -X user@server.com
```

Vital-it basics

Vital-it is a cluster. So a lot of people run jobs on it and you cannot just execute a job, you have to run it through a queue system (named LSF on vital-it). Basically, you write a script that you give to the queue system. When it can, it will run your script and send you the results when done. So never, ever, *EVER* run something directly on vital-it, always use the queue system.

Cmd	Usage	Description
man	man cmd	Show helps on a command, your best friend
bjobs	bjobs	Print the ID of jobs currently running
bsub	bsub < file	Submit a job to the queue system
bkill	bkill JOBID	Kill a job in the queue system using ID
bpeek	bpeek JOBID	Check output of a job