UNIX miscellaneous tips

Keyboard shortcuts

- Use tab to complete commands or filenames having partially typed them in
  e.g type `sam` then tab to complete this to the full command `samtools` or type `ls /ho` and then tab to complete to `ls /home`
  This is useful particularly with long filenames
- You can scroll back through previous commands you have typed using the up and down arrow keys. For example if you have just typed a very long command and pressed enter only to find that you made a mistake you can use the up arrow to select the previous command and edit it to correct the mistake and type enter again
- You can search for a previously typed command by pressing `ctrl-r` and then part of the command and it will search back through your history to find a command matching what you have typed
  e.g type `wget` and if you have typed this before you may see something like
  `(reverse-i-search)`wget': wget https://gitlab.com/cgps/ghru/pipelines/snp_phylogeny/raw/master/bin/filtered_bcf_to_fasta.py`

  You can then use the arrow keys to edit the command if needed and return to execute it.

Passwordless login using ssh keys

1. Generate a key on your local machine using
   a. `ssh-keygen` on a Linux or Mac machine (see here)
      This will generate a file called id_rsa.pub in the hidden .ssh directory in your home directory.
      Type `cat .ssh/id_rsa.pub` to show the contents which can then be shared with someone else. Never share the `.ssh/id_rsa` file which is the private key
   b. To do the same in Windows follow this tutorial to create it in Putty
2. Login in to the server with the username you would like passwordless access for example
   `ssh biouser@123.456.789.001`
   You will have to enter your password this time
   Add the key to a file called .ssh/authorized_keys. If it does not exist, first add it:
   You have to create the .ssh directory and the `authorized_keys` file the first time.
   Create the .ssh directory:
   `mkdir ~/.ssh`
   Set the right permissions:
   `chmod 700 ~/.ssh`
   Create the authorized_keys file:
   `touch ~/.ssh/authorized_keys`

Set the right permissions:
`chmod 600 ~/.ssh/authorized_keys`
The permissions are important! It won't work without the right permissions!

3. Now logout of the server and back in again. For example
   `ssh biouser@123.456.789.001`
   This time it will login you in without requiring a password since your private key in `.ssh/id_rsa` matches the public key contained in the `authorized_keys` file on the server

Aliases

Aliases are like shortcuts for long commands that save you typing in a frequently used command. Working with the CLI is all about doing things efficiently and bioinformaticians are lazy!

In your home directory (e.g /home/user1) there is a hidden file called `.bashrc` that is loaded every time you login to a terminal. This can perform many things such as

- define the way your command prompt looks
- change environmental variables such as PATH which is a list of directories that UNIX will look in for commands
- define aliases

To add a new alias open this file in an editor (for example vi)

`vi ~/.bashrc` **Note** ~ is a shortcut for your home directory

Go to the end of the file (press 'G' in vi) or go to a section of the file where other aliases are defined and add a new line and type the new alias. The format is

<SHORT CUT>='<LONG COMMAND'> for example to add a short cut for the command that lists all files in the directory newest file last (`ls -ltr`) the alias might be

```
z='ls -ltr'
```

Another example might be that if you are connecting to a server from a Linux or Mac machine and therefore not using PuTTY shared sessions you will often type something like `ssh ubuntu@123.456.789.001`. Rather than typing this each time make an alias such as

```
serv1='ssh ubuntu@123.456.789.001'
```

To activate the new aliases you can either logout and log back in again or reload the .bashrc file using the command `source .bashrc`

Finding files

Finding files in UNIX can be as simple or as sophisticated as you like. The general format is

```
find <STARTING DIRECTOR> <OPTIONS>
```

For example to just find a file with the name test.fas in the current directory you would type

```
find . -name test.fas
```

The name can include a wild card such as `*` to represent any character or a several possible characters can be specified within `{}`. For example to find all files ending in either _1.fastq.gz or _2.fastq.gz in /data type

```
find /data -name *_{1,2}.fastq.gz
```

To find only files or directories the `-type` argument can be used

The following command will find any directory from / downwards that contains data somewhere in the name

```
find / -type d -name *data*
```

The following command will find in the /data directory and all sub-directories any **file** whose name contains genome

```
find /data -type f -name *genome*
```

It is also possible to search for files that have been modified within a certain time
The following command will search the current directory and all subdirectories for files (including directories) modified within the last 2 days or the last 2 hours.

```
find . -mtime -2
find . -mmin -120m
```

Find files modified over 2 days ago

```
find . -mtime +2
```

These commands can be combined. For example the following command will search for files ending in .vcf that have been modified in the last 12 hours

```
find . -type f -name *.vcf -mmin -12h
```

Disk Usage

Your disk is getting full!! How do you find what is taking up the space.

Firstly to find how much disk space is being used on different partitions use the df (disk free) command with the -h argument to make the output sizes human-readable

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev             32G     0   32G   0% /dev
tmpfs           6.3G  908K  6.3G   1% /run
/dev/sda1       117G   87G   30G  75% /
tmpfs            32G     0   32G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs            32G     0   32G   0% /sys/fs/cgroup
/dev/sda15      105M  3.4M  102M   4% /boot/efi
/dev/sdb        984G  365G  569G  40% /data
```

In this case we can see that

- root partition has used 87G (75%) of disc space and has 30G free.
- the data partition has used 365G (40%) of disc space and has 569G free

To see what files are taking up the space we can use the du (disk usage) command again with the -h argument as well as -s to look at the usage by size. The following command will list the directories or files within /data and sort them by human-readable size

```
sudo du -sh /data/* | sort -h
```

```
....
2.1G  /data/mash_analysis
5.2G  /data/user5_data
7.6G  /data/user1_data
9.0G  /data/user3_data
13G   /data/contamination_files
15G   /data/user2_data
38G   /data/snp_analysis_tutorial
46G   /data/prodege
65G   /data/assembly
92G   /data/user4_data
```

Here we can see that user4_data is taking up the greatest space (92G). Looking further inside this directory we can find what subdirectories within `/data/user4_data` are taking up the most space

```
sudo du -sh /data/user4_data/analyses/*  | sort -h
```

```
51M   /data/user4_data/analyses/amr_analysis
60M   /data/user4_data/analyses/etec_abricate
41G   /data/user4_data/analyses/ETEC_assembly
52G   /data/user4_data/analyses/etec_snp_phylogeny
```