Molecular Microbiology 2024

Protocol 4

Logging in to remote course servers and uploading raw sequencing data

This protocol covers logging into a remote server using a secure shell (or "ssh") on MacOS or Windows.

Instructions for MacOS users, Windows Terminal Users, and VSCode users

1. Open **Terminal** (typically found in *Applications -> Utilities* on MacOS or from the Windows Start Menu, or by *Terminal -> New Terminal* in VSCode).

2. Log into the server with the following command, where `molmicroX` is your group username. You will then be prompted for your password - note that when you type your password it will not appear - this is to help keep it secret from people looking over your shoulder.

   ```
   ssh molmicroX@dnaseq1a.bio.au.dk
   ```

3. Use the `ls` command to list the files and folders present in your current directory (folder). The directory you first land in when you log in is your "home" directory. Type the following command and press enter.

   ```
   ls
   ```

4. We will now **m**a**k**e a new **dir**ectory called `genome` with the `mkdir` command - most of our work will be saved in the `genome` directory for this course. We will then **c**hange **d**irectory into the `genome` directory that we just created using the `cd` command.

   ```
   mkdir genome
   cd genome
   ```

5. **Only one group member needs to upload your data (steps 6–8)**. Type `ls` in the terminal - the `genome` directory you just created should be empty. Launch **CyberDuck** and click on *Open Connection* at the top. Select *SFTP (SSH File Transfer Protocol)* from the drop-down menu, enter `dnaseq1a.bio.au.dk` under *Server*, under *Username* put your username `molmicroX`, under *Password* enter your password, and under *SSH Private Key* select *None.* Click *Connect* when you're ready. If you receive a message about an "Unknown fingerprint" then just click *Allow.*

6. You should be able to see the `genome` folder that you just created. Double-click on it, then drag your sequencing data file into the Cyberduck window.

7. Go back to your open Terminal window and type `ls`. You should be able to see the files that you just uploaded.

8. Type `cd` to return to your home directory.

Linux Command Line Quick Reference

Changing directory (folder)

- `cd directory_name` - change into the directory called `directory_name`
- `cd ~` or `cd` - change into your home directory (from wherever you are)
- `cd ..` - change into the parent directory of your current directory (i.e. one level up)

Listing directory contents

- `ls` - list all files in the current directory
- `ls directory_name` - list all files in the directory name
- `ls -lh` - list all files including extra information (including file sizes)
- `ls *.fasta` - list all files with filenames ending in `.fasta` (the wildcard, `*`, can be used in any command as a way of referring to several different files with a common beginning or ending)

Copying and moving files

- `mkdir directory_name` - make a new directory inside the current directory called `directory_name`
- `cp filename directory_name/` - make a copy of the file `filename` inside `directory_name`
- `cp filename new_filename` - make a copy of the file `filename` called `new_filename`
- `mv filename new_filename` - rename file `filename` to `new_filename`
- `mv filename directory_name/` - move file `filename` into `directory_name`

Removing (deleting) files

- `rm filename` - delete the file called `filename` (warning, there's no recycle bin! Once you delete it it's gone forever, so be careful!)
- `rm -r directory_name` - delete the directory called `directory_name` (see note above about there being no recycle bin)

Unzipping files

- `gunzip filename` - unzip the gzipped file called `filename`
- `gunzip *.gz` - unzip all gzipped files with filenames ending in `.gz`

Viewing files

- `cat filename` - print out the contents of `filename` to the command line
- `less filename` - print out the contents of `filename` to the command line in a format that lets you scroll slowly through the file with the up/down arrow keys. Type `/searchterm` to search for `searchterm` in a file, press `q` to quit.
- `head filename` - view the first ten lines of a file (useful for investigating files too large to display)
- `tail filename`- view the last ten lines of a file

- `fgrep searchterm filename` - display all lines in file `filename` that contain `searchterm`

Getting help

- `man commandname` - view the manual page for a given command, e.g. `man ls`. Type `q` to quit `man`.
- `commandname -h` or `commandname --help` or `commandname -help` - get help for a command that there's no manual page for (this typically works but not always)

Other tips

- **Autocomplete** - Start typing the name of a file then press `tab` on your keyboard - if what you've typed so far is specific enough to identify that file, then it will autocomplete the whole file, or at least as much as it can guess.
- **Up/down arrows** - Display the last command that you entered by pressing "up", keep pressing "up" to keep going back through your history, press "down" to go forward in time.
- **Piping** - The `|` character is a pipeline that takes the output from one command and "pipes" it as input into another command
- **Ctrl+C** - Terminate running programs by holding down the "Ctrl" key on your keyboard and hitting "C". Kind of like using "Force Quit"/"Ctrl+Alt+Del" then stopping a process on MacOS/Windows.

History

- `history` - print all the commands you've given to the server. If you just want to see the most recent ones, this is a good one to combine with the tail command `history | tail`. It's also handy to combine this with `fgrep` to see the last time you used a command. For example, did you copy a file somewhere but can't remember where? Then use `history | fgrep cp` to see the most recent uses of the `cp` command.