# Getting Started

The data files required for this workshop are located in a public directory of the instructor’s account. You need to have this in your home directory (/home/username) before you start the exercise 1. You can download the compressed directory and extract it by simple commands given below. You will learn about these commands later in the exercise.

Open the terminal and enter these commands (commands are case sensitive) and each command should be entered in a single line followed by ⮠ (Enter) key

wget http://gif.biotech.iastate.edu/WORKSHOP\_FILES.tar.gz ⮠

tar –xvzf WORKSHOP\_FILES.tar.gz ⮠

Once your cursor (command prompt) comes back to the original position, type

ls ⮠

You should see WORKSHOP\_FILES listed there.

# UNIX Exercise 1

**This exercise is designed to provide the basic skills required for working in the UNIX environment, using plenty of relevant examples, specifically for biologists. If you are using your personal computer, make sure that you have downloaded the files required for the workshop. This exercise will provide you information regarding navigation, files and directory creation/modification and some administrative things related to file permissions.**

## **Naviagation**

This section will introduce you to some basic file/directory navigation and manipulation techniques.

### To know the present location of your command

pwd

/home/username

Returns you the present working directory (*p*rint *w*orking *d*irectory)

This means, you are now working in the username directory, which is located in home directory. The directory that you will be in after logging in is your home directory. You can also avoid writing the full path by using ~ in front of your username.

~username same as /home/username

Present directory is represented as . (dot) and parent directory is represented as .. (dot dot)

### Changing directories

To jump from one directory to another we use the cd (*c*hange *d*irectory) command.

cd ..

Changes your present location to the parent directory

cd DIRECTORY

This changes your location back to your DIRECTORY.

**Task 1.1: Now change your directory to the** WORKSHOP\_FILES **directory present in your home directory.**

**NOTE:** You can type in first few letters of the directory name and then press tab to auto complete rest of the name (especially useful when the file/directory name is long). This only works when there are unique matches for the starting letters you have typed. If there is more than one matching files/directories pressing tab twice will list all the matching names. You can also recall your previous commands by pressing up/down arrow or browse all your previously used commands by typing history on your terminal (typically last 500 commands will be saved in this file).

## **Directories and Files**

### Making directories

To create a directory, mkdir (*m*a*k*e *dir*ectory) can be used.

mkdir DIRECTORY

Unlike PC/Mac folders, here you can’t have space in your directory name. Alternatively, you can also specify the path where you want to create your new folder.

**Task 1.2: Make a new directory named** FirstDirectory **within the** WORKSHOP\_FILES **directory. Then change your directory to the** FirstDirectory**.**

mkdir FirstDirectory

### Copying Directories

To copy a file, cp (*c*o*p*y) command is used. When using this command you have to provide both source file and destination file.

cp SOURCE DESTINATION

You can also specify the absolute path of the source and/or destination file. To know more about any command you can use man command, which opens the manual of the command you ask.

man cp

This opens the manual for the cp command. Take a look at the manual of cp command (use arrow keys to move top or bottom of the page). OPTIONS are optional parameters that can be used to accomplish more from the same command. *Eg.,* by using option –i with the regular cp command, you can always make sure that you are not overwriting the existing file while copying. The syntax for using the options will also be provided in the manual. **To exit, press** q.

*Looking at the man page for cp command, what options can be used to copy a directory (including all files within it)? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*How else you can get help on cp command (other than ‘man’)? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Task 1.3: Now change your directory back to the home directory. Create a copy of** WORKSHOP\_FILES **and name it as** BACKUP\_WORKSHOP**).** This will serve as a backup copy of all files that are required for the workshop (in case you accidentally modify the contents while working).

cp -r WORKSHOP\_FILES BACKUP\_WORKSHOP

### Moving Directories

To move a file or a directory, mv (*m*o*v*e) command is used. Again, like the cp command you need to provide both source file and destination file.

mv SOURCE DESTINATION

Absolute path also works fine. Some of the options used by cp command also work with mv command. mv can also be used to rename files and directories

mv OLDNAME NEWNAME

**Task 1.4: Rename WORKSHOP\_FILES as tutorials.**

mv WORKSHOP\_FILES tutorials

### Viewing the contents of the directory

The contents of a dir can be viewed using ls (list) command.

ls DIRECTORY *# now try it with* tutorials *directory*

If no directory name is provided then ls will list all the contents of the present directory.

Like any other command, you can use absolute path or abbreviated path. There are also various options available for ls command.

Some very useful options include:

ls –l

Lists all the files in lengthy or detailed view

ls –t

Lists all the files, sorted based on creation time

ls –S

Lists all the files, sorted based on size

You can also combine these options together for getting more focused results.

*Looking at the manual for ls, what option can you use to view hidden files in a directory (files starting with dot)? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Can you sort the files based on its extension? How? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**Task 1.5: Examine the contents of the** tutorials **directory. Try options such as** -l**,** -t**,** -a **and** -X**. Also check if you can combine many options together (like** -la **or** -lh **etc).** Try these:

ls -l tutorials

ls -a

ls -1 tutorials

ls -lh tutorials

ls -t tutorials

### Creating and editing files

touch FILENAME

Creates a new file in the present location

nano FILENAME

Like notepad, this text editor lets you edit a file.

**Task 1.6: Create a new file named** firstfile **inside the** tutorials **directory. You can create using touch or using** nano**. Then add some contents (Your name and email address) to the** firstfile **(using** nano**). After editing, press** Ctrl + X **to exit, then enter** y **to save changes and confirm the file name**.

touch firstfile

nano firstfile

### Viewing contents of the files

There are various commands to print the contents of the file in bash. Most of these commands are often used in specific contexts. All these commands when executed with filenames displays the contents on the screen. Most common ones are less, more, cat, head and tail.

less FILENAME *try this:* less AT\_cDNA.fa

Displays file contents on the screen with line scrolling (to scroll you can use arrow keys, PgUp/PgDn keys, space bar or Enter key). **When you are done press** q **to exit**.

more FILENAME *try this:* more AT\_cDNA.fa

Like less command, also, displays file contents on the screen with line scrolling but uses only space bar or Enter key to scroll. **When you are done press** q **to exit**.

cat FILENAME *try this:* cat AT\_cDNA.fa

Simplest form of displaying contents. It *cat*alogs the entire contents of the file on the screen. In case of large files, entire file will scroll on the screen without pausing

head FILENAME *try this:* head AT\_cDNA.fa

Displays only the starting lines of a file. The default is first ten lines. But, any number of lines can be displayed using –n option (followed by required number of lines).

tail FILENAME *try this:* tail AT\_cDNA.fa

Similar to head, but displays the last 10 lines. Again –n option can be used to change this.

More information about any of these commands can be found in man pages (man command)

**Task 1.7: Try using all these commands on the** RefSeq.faa**. You are also welcome to try these commands on various other files that are present in the tutorials directory.** These commands don’t change the contents of the file; they just display them on the screen.

### Deleting files and directories

To delete directories from the system, you can use rmdir (*r*e*m*ove *dir*ectory) command. You can also use rm command to delete file(s).

rmdir DIRECTORY

The directory should be empty before you use the rmdir command.

rm FILE

To delete a file rm command can be used

Some useful options include

–r recursively delete files

-f delete forcefully

rm –rf DIRECTORY **[DO NOT USE THIS NOW!]**

When you want to delete a folder, with all its content

**Task 1.8: Delete the directory named** delete\_me **inside the** tutorials **directory (to do this you may first want to delete the** sample.txt **file inside this directory).**

cd delete\_me

rm sample.txt

cd ..

rmdir delete\_me

### compressing files

There are several options for archiving and compressing groups of files or directories. Compressed files are not only easier to handle (copy/move) but also occupy less size on the disk (less than 1/3 of the original size). In Linux systems you can use zip, tar or gz for archiving and compressing files/directories.

ZIP compression/extraction

zip OUTFILE.zip INFILE.txt

Compress INFILE.txt

zip -r OUTDIR.zip DIRECTORY

Compress all files in a DIRECTORY into one archive file (OUTDIR.zip)

zip -r OUTFILE.zip . -i \*.txt

Compress all txt files in a DIRECTORY into one archive file (OUTFILE.zip)

unzip SOMEFILE.zip

Decompress a file

**Task 1.9: Zip** AT\_genes.gff **file located in the** tutorials **directory. Check the file size before and after zip compression (Hint: use** ls –lh **to check file sizes).**

zip AT\_genes.gff.zip AT\_genes.gff

*Is there any size difference before and after compressing? Y/N*

tar (*t*ape *ar*chive) utility saves many files together into a single archive file, and restores individual files from the archive. It also includes automatic archive compression/decompression options and special features for incremental and full backups.

tar -cvf OUTFILE.tar INFILE

archive INFILE

tar -czvf OUTFILE.tar.gz INFILE

archive and compress file INFILE

tar -tvf SOMEFILE.tar

list contents of archive SOMEFILE.tar

tar -xvf SOMEFILE.tar

extract contents of SOMEFILE.tar

tar -xzvf SOMEFILE.tar.gz

extract contents of gzipped archive SOMEFILE.tar.gz

tar -czvf OUTFILE.tar.gz DIRECTORY

archive and compress all files in a directory into one archive file

tar -czvf OUTFILE.tar.gz \*.txt

archive and compress all ".txt" files in current directory into one archive file

**Task 1.10: Archive and compress the** BACKUP\_WORKSHOP **directory you created in Task 1.3 (you can name it as** backup.tar.gz **or anything you want)**

tar -czvf backup.tar.gz BACKUP\_WORKSHOP

gzip (*g*nu *zip*) compression utility designed as a replacement for compress, with much better compression and no patented algorithms. The standard compression system for all GNU software.

gzip SOMEFILE

compress SOMEFILE (also removes uncompressed file)

gunzip SOMEFILE.gz

uncompress SOMEFILE.gz (also removes compressed file)

**Task 1.11:** gzip **the file** AT\_genes.gff **and examine the size.** gunzip **it back so that you can use this file for the later exercises.**

gzip AT\_genes.gff

ls -lh

gunzip AT\_genes.gff.gz

ls –lh

## Administrative Commands

### Changing permissions

All files in the UNIX system will have a set of permissions which define what can be done with that file and by whom. (What = read (view contents), write (modify) and execute (run script) Whom=User (owner), group (that account belongs to) and everyone else). They are denoted as

**PERMISSIONS RELATIONS**

read r owner u

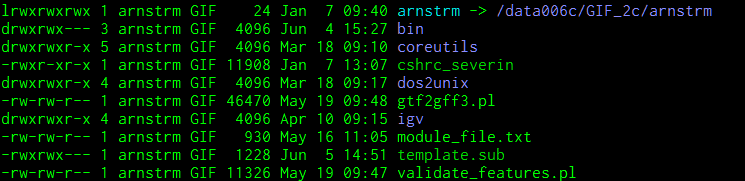
write w group g

execute x others o

all users a

To look at the permissions for any file, you can list the files with l option (ls –l).

Permissions User Group Size Date modified Name



u g o

(d=directory, l=link, r=read, w=write, x=execute, -=blank, u=user, g=group, o=others)

To set/modify a file's permissions you need to use the chmod command (*ch*ange *mod*e). Only the owner of a file can alter a file's permissions. The syntax:

chmod [OPTIONS] RELATIONS[+ or -]PERMISSIONS FILE

Add permissions

chmod RELATIONS+PERMISSIONS FILENAME

chmod g+rwx FILENAME grants read, write and execute permissions for group

chmod g+r FILENAME grants read permission for group

chmod a+rwx FILENAME makes the file public (don’t do this to any file/directory unless you want to share)

Remove permissions

chmod RELATIONS-PERMISSIONS FILENAME

chmod g-wx FILENAME removes write and execute permissions for group

chmod g-rwx FILENAME removes all permissions for group

chmod a-rwx FILENAME removes all permissions for others

chmod a-x FILENAME removes execution permissions for others

OPTIONS include

-R recursively (the permissions are applied to all the files, directories present inside the directory)

**Task 1.12: Check the permissions for the files located in the** tutorials **directory. Do**

ls -l

*What permissions does the group have on these files? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Which group does your account belong to? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*