

# Wild Cards & Nano

Dr. Princess Rodriguez

2025-01-22

# Learning Objectives

- Learn to use tab completion to simplify typing file and directory paths.
- Use of the asterisk \* wildcard to match and select multiple files in a directory.
- View the contents of a file using commands such as `less`, `head`, or `cat`.
- Use Nano, a simple text editor, to create and edit files directly from terminal.

## Class Exercise

Please complete the following class activity below.

[Class-activity] (<https://forms.gle/6L5eDqiiLijc8i6H8>)

# Wild cards

The \* or wildcard character is a shortcut for “everything”.

You can use the \* by typing in shift + 8.

```
ls *.fq
```

# Command History

You can easily access previous commands by hitting the up arrow key on your keyboard, this way you can step backwards through your command history. On the other hand, the down arrow key takes you forward in the command history.

```
history
```

The wildcard \* can be placed anywhere in your pattern. For example:

```
ls Mov10*fq
```

## So how does \* actually work?

The Shell (bash) considers an asterisk \* to be a wildcard character that can match one or more occurrences of any character, including no character. In the example above the \* took place of 13 characters!

# The “?” wildcard

Another wildcard that is sometimes helpful is ?

- ? is similar to \* except that it is a placeholder for exactly one position.
- Recall that \* can represent any number of following positions, including no positions.



# cat command

The easiest way to examine a file is to print out all of its contents using the command `cat`. We can test this out by printing the contents of `~/unit1_unix/other/sequences.fa`

```
cat sequences.fa
```

*cat stands for concatenate; it has many uses and printing the contents of a file onto the terminal is one of them.*

# less command

```
less Mov10_oe_1.subset.fq
```

Rather than printing to screen, the `less` command opens the file in a new buffer allowing you to navigate through it.

| key   | action                             |
|-------|------------------------------------|
| SPACE | to go forward                      |
| b     | to go backwards                    |
| g     | to go to the beginning of the file |
| G     | to go to the end of a file         |
| q     | to quit or exit less               |

# head and tail commands

The commands are `head` and `tail` and they just let you look at the beginning and end of a file respectively.

```
head Mov10_oe_1.subset.fq
```

```
tail Mov10_oe_1.subset.fq
```

# Good names for files and directories

- 1 Don't begin the name with - (dash).
- 2 Don't begin the name with numbers.
- 3 Stick with letters in the beginning and then use numbers, . (period), - (dash), or an \_ (underscore) in the middle of the file or directory name.

# Creating or editing a file with Nano

To create a new file or edit an existing one type:

```
nano filename
```

## Citation

*This lesson has been developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC). These are open access materials distributed under the terms of the Creative Commons Attribution license (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

- *The materials used in this lesson were derived from work that is Copyright © Data Carpentry (<http://datacarpentry.org/>). All Data Carpentry instructional material is made available under the Creative Commons Attribution license (CC BY 4.0).*
- *Adapted from the lesson by Tracy Teal. Original contributors: Paul Wilson, Milad Fatenejad, Sasha Wood and Radhika Khetani for Software Carpentry (<http://software-carpentry.org/>)*
- *Other Authors include: Sheldon McKay, Mary Piper, Radhika Khetani, Meeta Mistry, Jihe Liu, Mary Piper, Meeta Mistry, Jihe Liu, & Will Gammerdinger*