Introduction to the Command Line for Genomics (../) (../01introduction/index.html)

(../03worki withfiles/i

Navigating Files and Directories

Overview

Teaching: 30 min Exercises: 20 min Questions

- How can I perform operations on files outside of my working directory?
- What are some navigational shortcuts I can use to make my work more efficient?

Objectives

- Use a single command to navigate multiple steps in your directory structure, including moving backwards (one level up).
- Perform operations on files in directories outside your working directory.
- Work with hidden directories and hidden files.
- Interconvert between absolute and relative paths.
- Employ navigational shortcuts to move around your file system.

Moving around the file system

We've learned how to use pwd to find our current location within our file system. We've also learned how to use cd to change locations and 1s to list the contents of a directory. Now we're going to learn some additional commands for moving around within our file system.

Use the commands we've learned so far to navigate to the shell_data/untrimmed_fastq directory, if you're not already there.

Bash

\$ cc

\$ cd shell_data

\$ cd untrimmed_fastq

What if we want to move back up and out of this directory and to our top level directory? Can we type cd_shell_data? Try it and see what happens.

Bash

\$ cd shell_data

Output

-bash: cd: shell_data: No such file or directory

Your computer looked for a directory or file called shell_data within the directory you were already in. It didn't know you wanted to look at a directory level above the one you were located in.

We have a special command to tell the computer to move us back or up one directory level.

Bash

\$ cd ..

Now we can use pwd to make sure that we are in the directory we intended to navigate to, and 1s to check that the contents of the directory are correct.

Bash

\$ pwd

Output

/home/dcuser/shell_data

Bash

\$ 1s

Output

sra_metadata untrimmed_fastq

From this output, we can see that .. did indeed take us back one level in our file system.

You can chain these together like so:

Bash

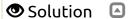
\$ 1s ../../

prints the contents of /home, which is one level up from your root directory.

Finding hidden directories

First navigate to the shell_data directory. There is a hidden directory within this directory. Explore the options for 1s to find out how to see hidden directories. List the contents of the directory and identify the name of the text file in that directory.

Hint: hidden files and folders in Unix start with . , for example .my_hidden_directory



First use the man command to look at the options for 1s.

Bash

\$ man ls

The -a option is short for all and says that it causes 1s to "not ignore entries starting with." This is the option we want.

Bash

\$ 1s -a

Output

. .. .hidden sra_metadata untrimmed_fastq

The name of the hidden directory is $\ .$ hidden $\ .$ We can navigate to that directory using $\ cd \ .$

Bash

\$ cd .hidden

And then list the contents of the directory using 1s.

Bash

\$ 1s

Output

youfoundit.txt

The name of the text file is youfoundit.txt.

Examining the contents of other directories

By default, the 1s commands lists the contents of the working directory (i.e. the directory you are in). You can always find the directory you are in using the pwd command. However, you can also give 1s the names of other directories to view. Navigate to your home directory if you are not already there.



Then enter the command:

Bash \$ ls shell_data

```
Output
sra_metadata untrimmed_fastq
```

This will list the contents of the shell_data directory without you needing to navigate there.

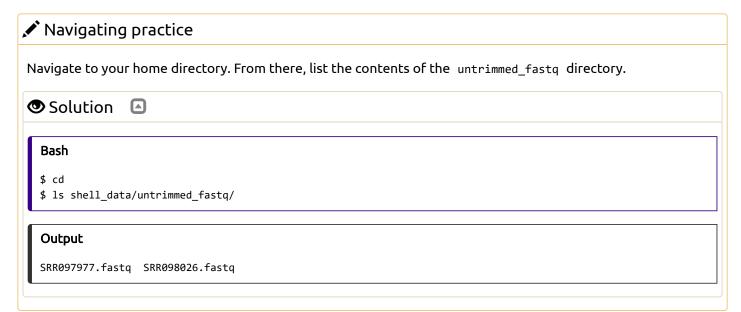
The cd command works in a similar way.

Try entering:

```
Bash

$ cd
$ cd shell_data/untrimmed_fastq
```

This will take you to the untrimmed_fastq directory without having to go through the intermediate directory.



Full vs. Relative Paths

The cd command takes an argument which is a directory name. Directories can be specified using either a *relative* path or a full *absolute* path. The directories on the computer are arranged into a hierarchy. The full path tells you where a directory is in that hierarchy. Navigate to the home directory, then enter the pwd command.

Bash \$ cd \$ pwd

You will see:

Output

/home/dcuser

This is the full name of your home directory. This tells you that you are in a directory called dcuser, which sits inside a directory called home which sits inside the very top directory in the hierarchy. The very top of the hierarchy is a directory called / which is usually referred to as the *root directory*. So, to summarize: dcuser is a directory in home which is a directory in /.

Now enter the following command:

Bash

\$ cd /home/dcuser/shell_data/.hidden

This jumps forward multiple levels to the .hidden directory. Now go back to the home directory.

Bash

\$ cd

You can also navigate to the .hidden directory using:

Bash

\$ cd shell_data/.hidden

These two commands have the same effect, they both take us to the .hidden directory. The first uses the absolute path, giving the full address from the home directory. The second uses a relative path, giving only the address from the working directory. A full path always starts with a / . A relative path does not.

A relative path is like getting directions from someone on the street. They tell you to "go right at the stop sign, and then turn left on Main Street". That works great if you're standing there together, but not so well if you're trying to tell someone how to get there from another country. A full path is like GPS coordinates. It tells you exactly where something is no matter where you are right now.

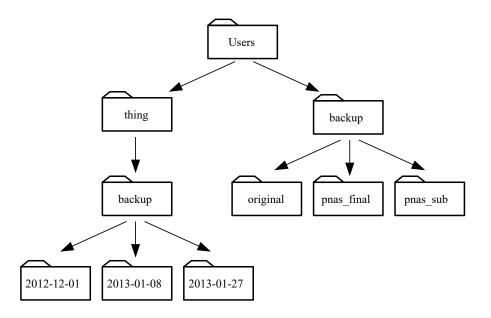
You can usually use either a full path or a relative path depending on what is most convenient. If we are in the home directory, it is more convenient to enter the relative path since it involves less typing.

Over time, it will become easier for you to keep a mental note of the structure of the directories that you are using and how to quickly navigate amongst them.

Relative path resolution

Using the filesystem diagram below, if pwd displays /Users/thing, what will 1s ../backup display?

- 1. ../backup: No such file or directory
- 2. 2012-12-01 2013-01-08 2013-01-27
- 3. 2012-12-01/ 2013-01-08/ 2013-01-27/
- 4. original pnas_final pnas_sub



Solution

- 1. No: there is a directory backup in /Users.
- 2. No: this is the content of Users/thing/backup, but with .. we asked for one level further up.
- 3. No: see previous explanation. Also, we did not specify -F to display / at the end of the directory names.
- 4. Yes: ../backup refers to /Users/backup.

Navigational Shortcuts

There are some shortcuts which you should know about. Dealing with the home directory is very common. The tilde character, ~ , is a shortcut for your home directory. Navigate to the shell_data directory:

Bash \$ cd \$ cd shell_data

Then enter the command:

Bash	
\$ 1s ~	

Output

R r_data shell_data

This prints the contents of your home directory, without you needing to type the full path.

The commands cd, and cd ~ are very useful for quickly navigating back to your home directory. We will be using the ~ character in later lessons to specify our home directory.

Key Points

- The /, \sim , and \ldots characters represent important navigational shortcuts.
- Hidden files and directories start with . and can be viewed using 1s -a.
- Relative paths specify a location starting from the current location, while absolute paths specify a location from the root of the file system.

```
(../01-
introduction/index.html)
```

) (../03worki withfiles/i

Licensed under CC-BY 4.0 () 2018–2019 by The Carpentries () Licensed under CC-BY 4.0 () 2016–2018 by Data Carpentry (http://datacarpentry.org)

Edit on GitHub (https://github.com/datacarpentry/shell-genomics/edit/gh-pages/_episodes/02-the-filesystem.md) / Contributing (https://github.com/datacarpentry/shell-genomics/blob/gh-pages/CONTRIBUTING.md) / Source (https://github.com/datacarpentry/shell-genomics/) / Cite (https://github.com/datacarpentry/shell-genomics/blob/gh-pages/CITATION) / Contact (mailto:)

Using The Carpentries theme (https://github.com/carpentries/carpentries-theme/) — Site last built on: 2019-11-23 19:34:47 +0000.