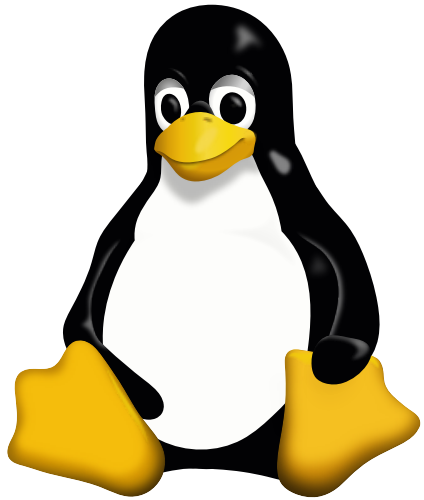
Leiden University
Medical Center

Practical Linux

Core 2



Department of Human Genetics

Outline

- Manipulate files and directories
- Inspect files
- Edit files
- Disk usage
- File name expansion
- Search and count in/for files
- Output redirection

Manipulate Files and Directories

`mkdir [option].. [directory]...`

Create new `directories`.

`cp [option]... [source]... [destination]`

Copies the `source` to the `destination`.

`mv [option]... [source]... [destination]`

Move or rename `source` to `destination`.

`rm [options]... [file/directory]...`

Remove specified `files` or `directories`.

Manipulate Files and Directories

`mkdir [option].. [directory]...`

Create new `directories`.

`cp [option]... [source]... [destination]`

Copies the `source` to the `destination`.

`mv [option]... [source]... [destination]`

Move or rename `source` to `destination`.

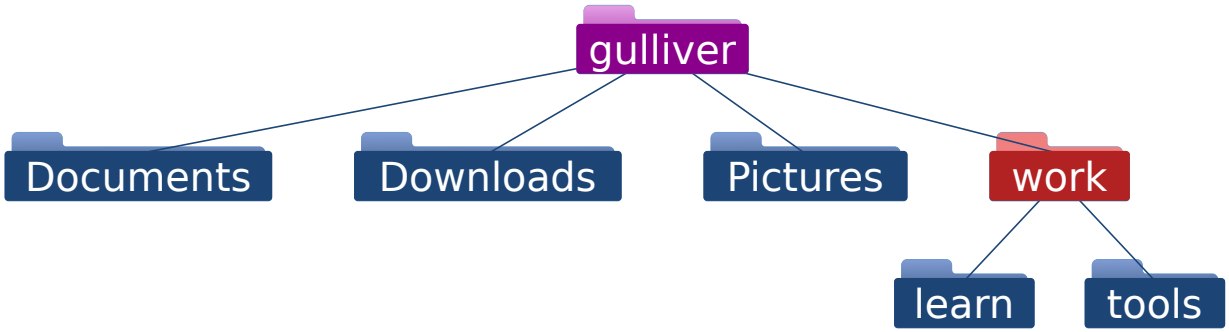
`rm [options]... [file/directory]...`

Remove specified `files` or `directories`.

Removing, i.e., deleting, erasing, is forever - no trash bin.

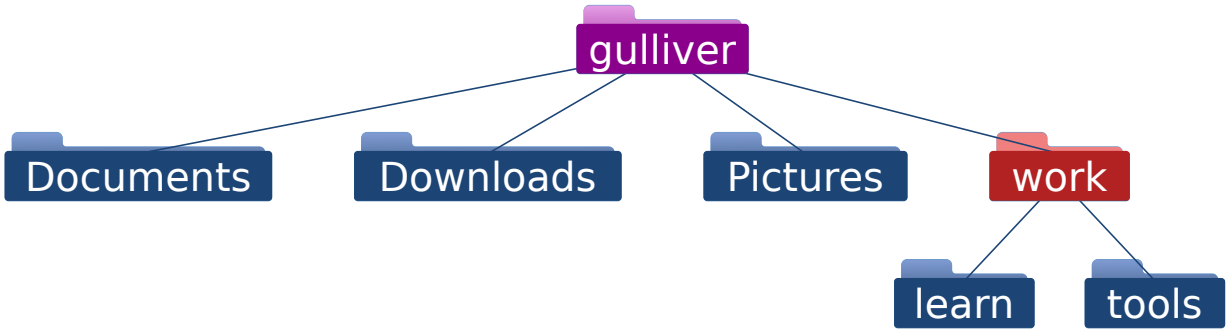
Create directories

```
$ mkdir
```



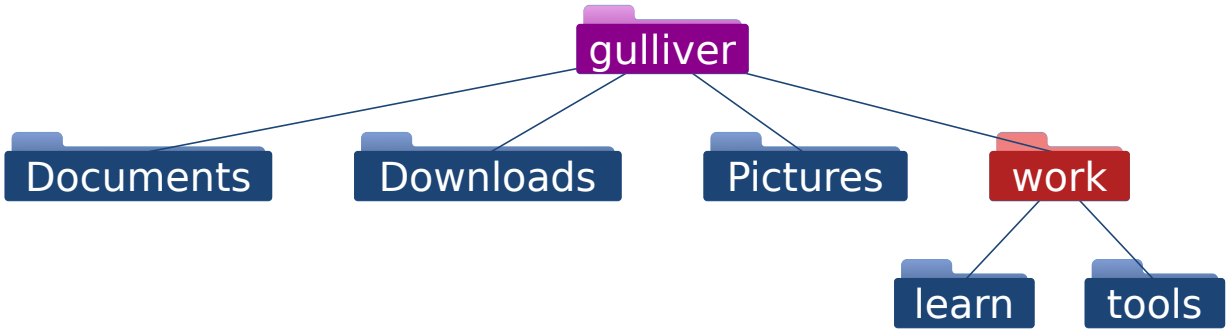
Create directories

```
$ mkdir project_x
```



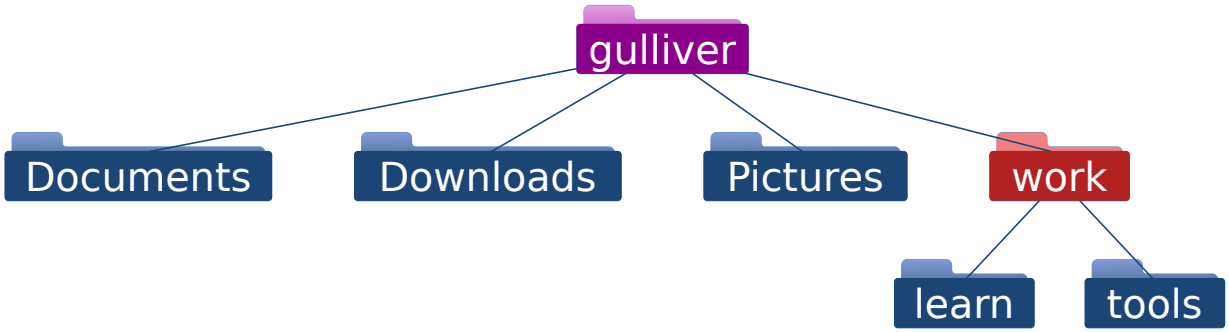
Create directories

```
$ mkdir project_x
$
```



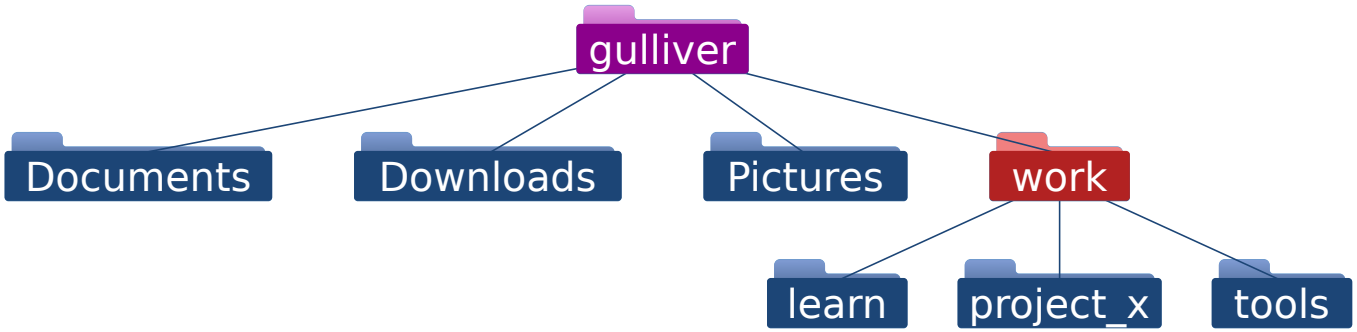
Create directories

```
$ mkdir project_x
$ ls
```



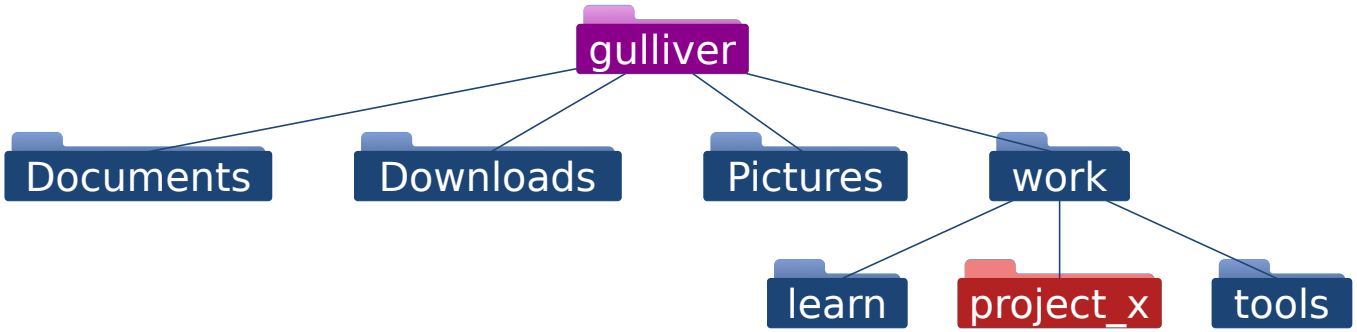
Create directories

```
$ mkdir project_x
$ ls
learn  project_x  tools
$
```



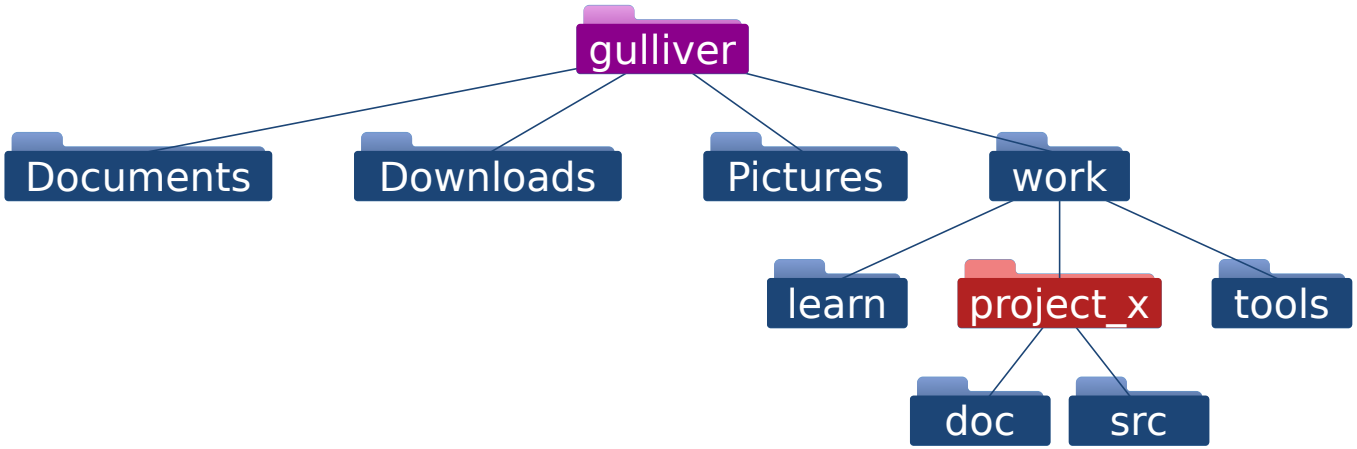
Create directories

```
$ mkdir data/human doc src
mkdir: cannot create directory 'data/human': No such file
or directory
$
```



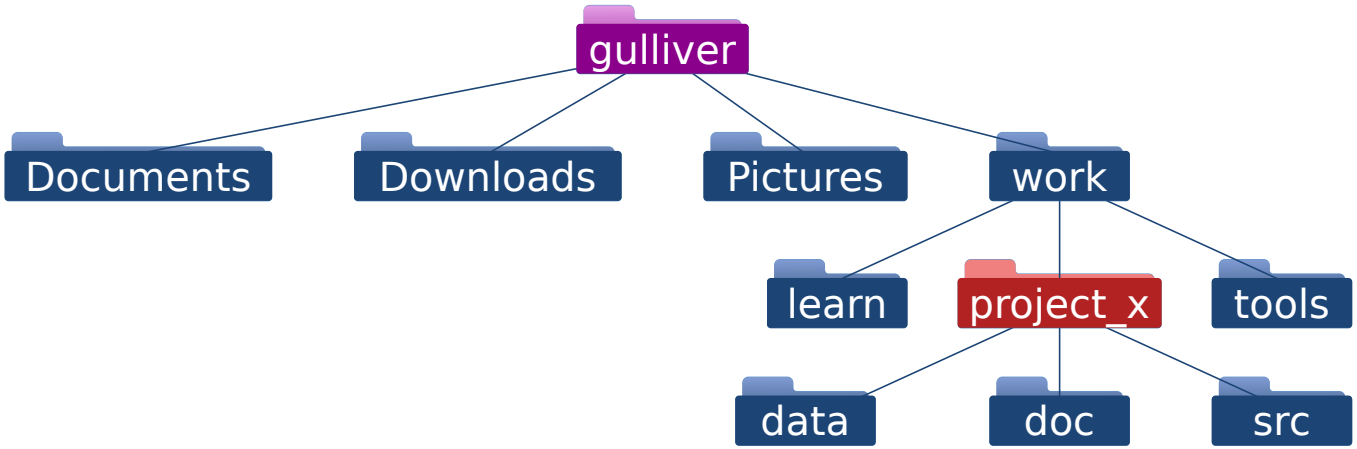
Create directories

```
$ ls
doc  src
$
```



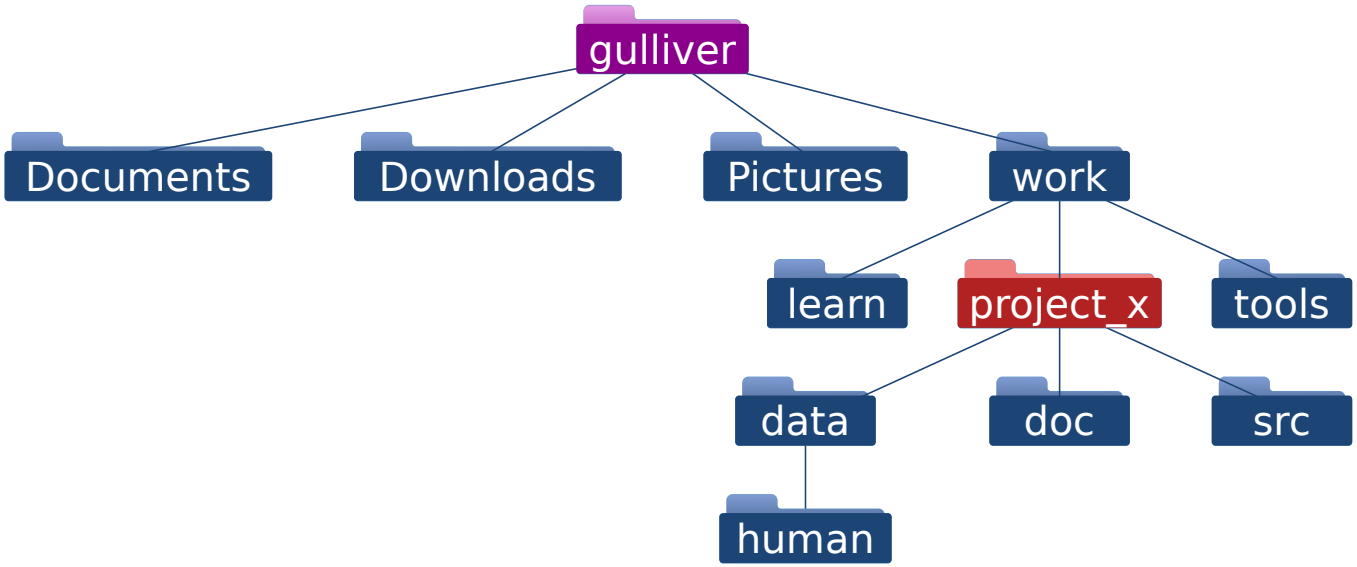
Create directories

```
$ mkdir -p data/human
$ ls
data  doc  src
$
```



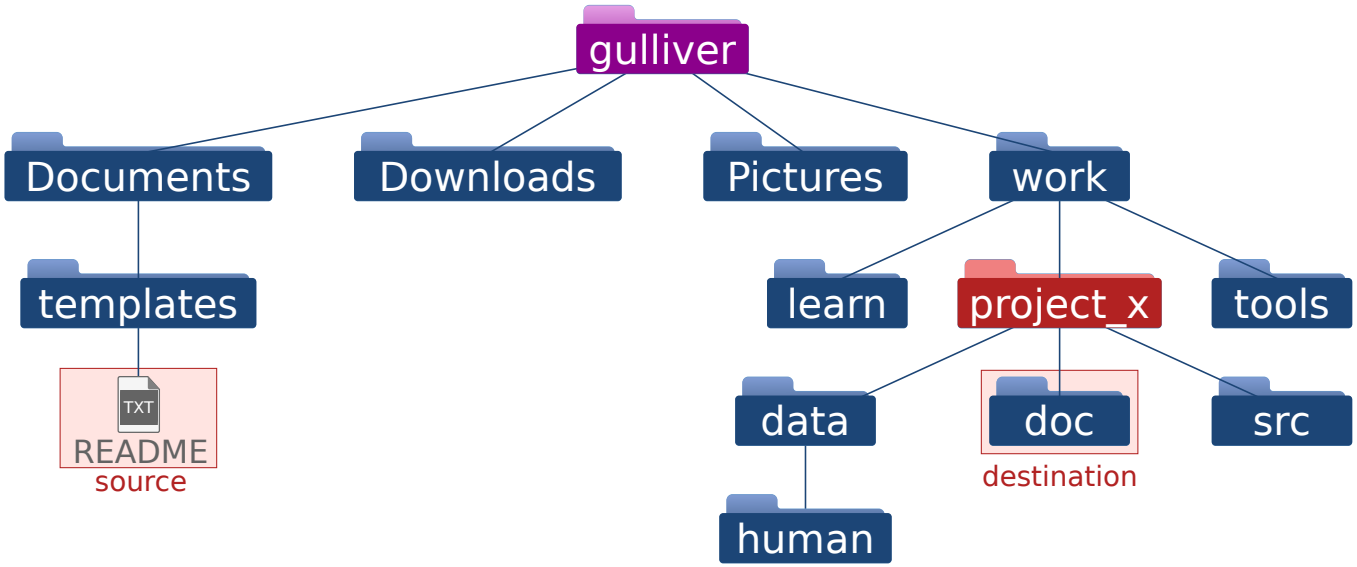
Create directories

```
$ ls data
human
$
```



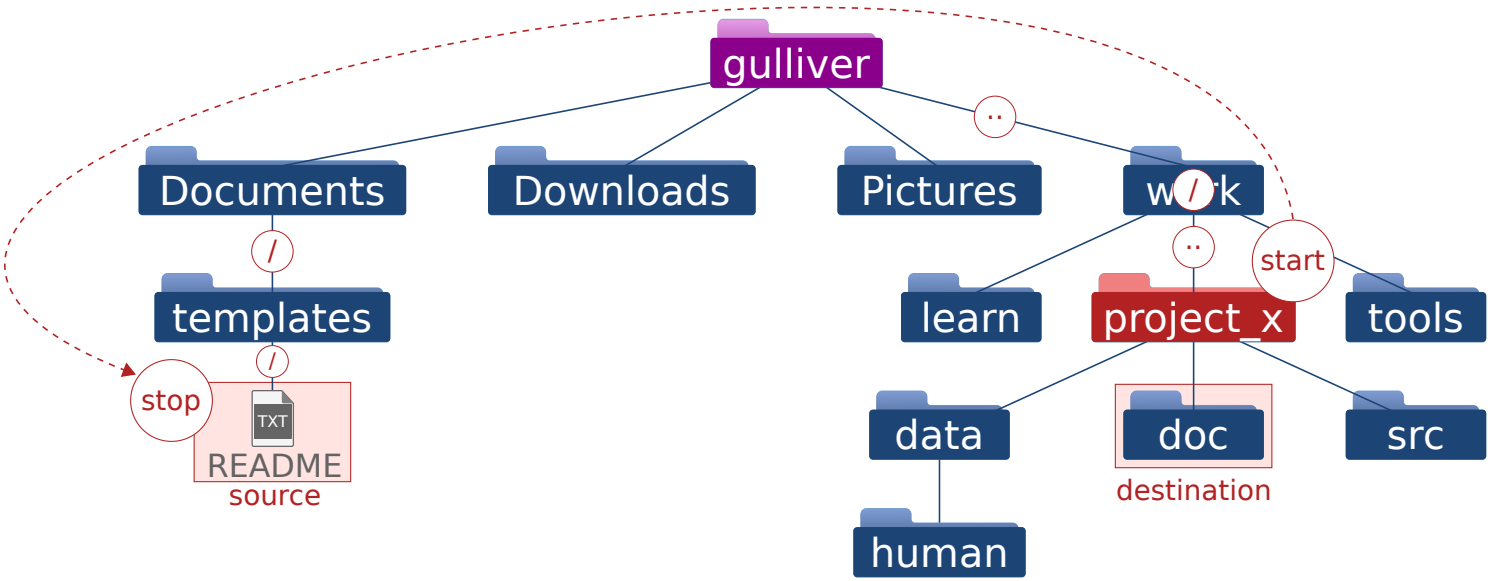
Copy files

```
$ cp
```



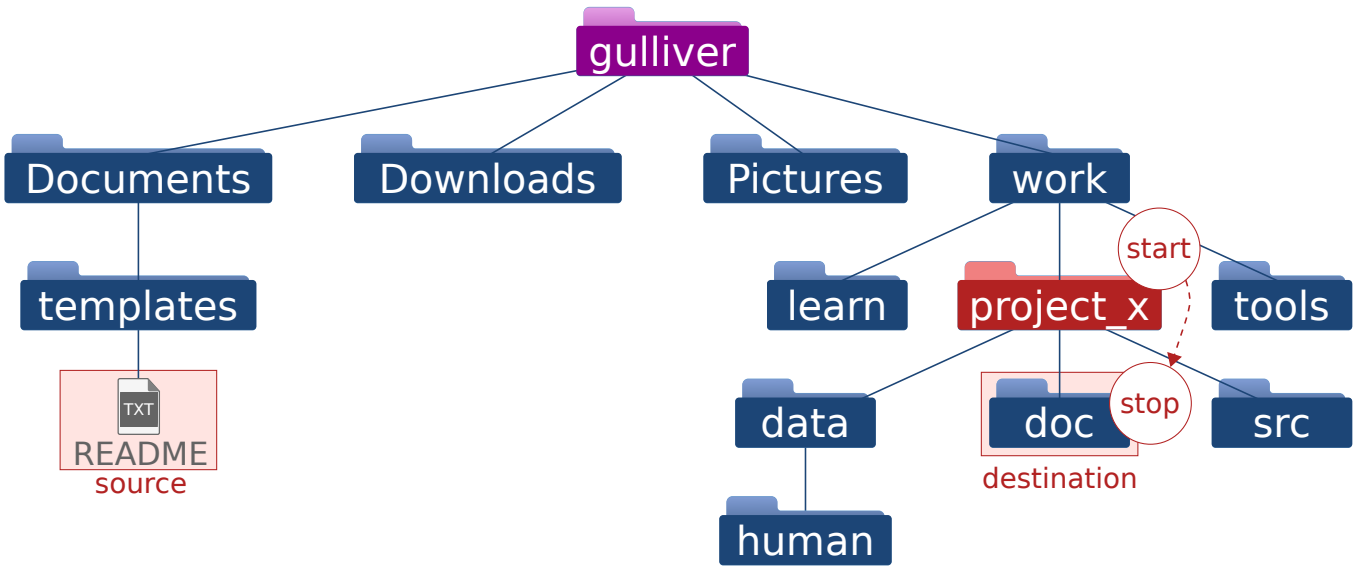
Copy files

```
$ cp ../../Documents/templates/README
```



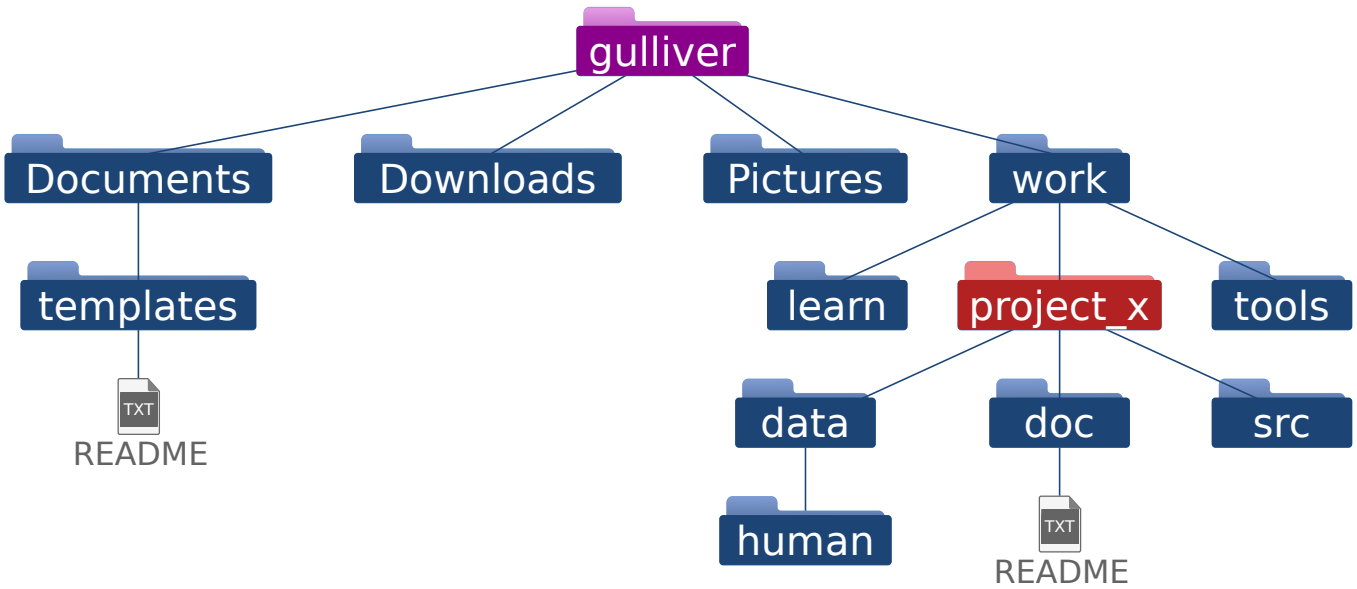
Copy files

```
$ cp ../../Documents/templates/README doc
```



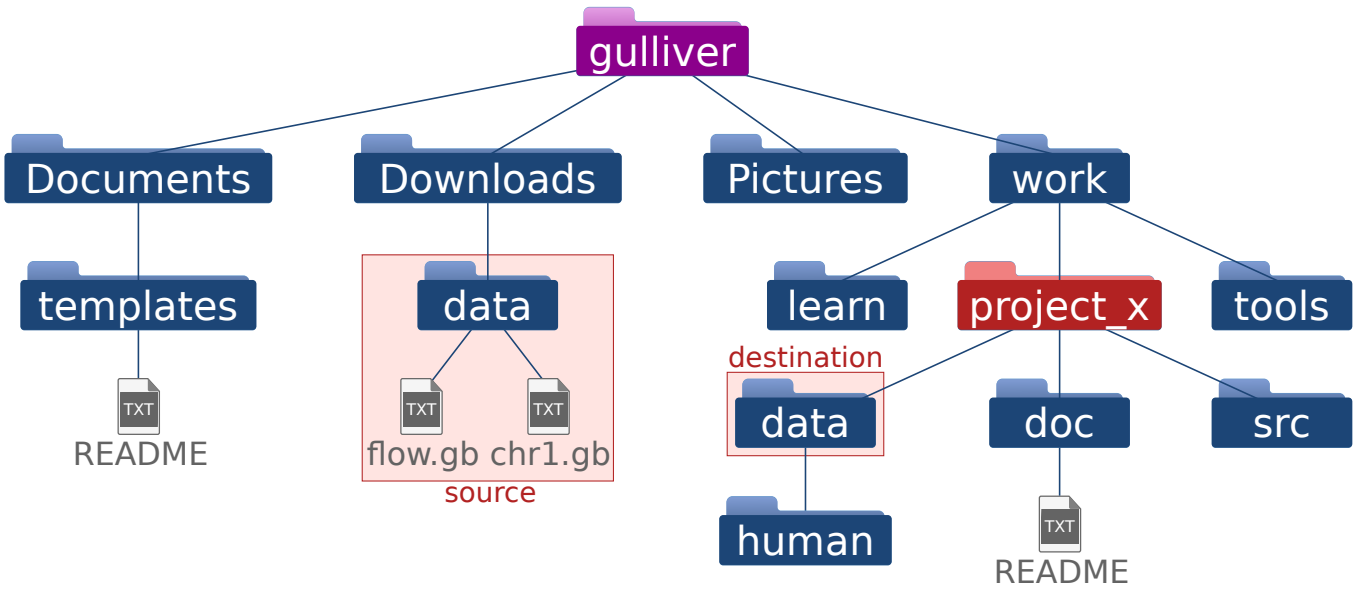
Copy files

```
$ cp ../../Documents/templates/README doc
$ ls doc
README
$
```



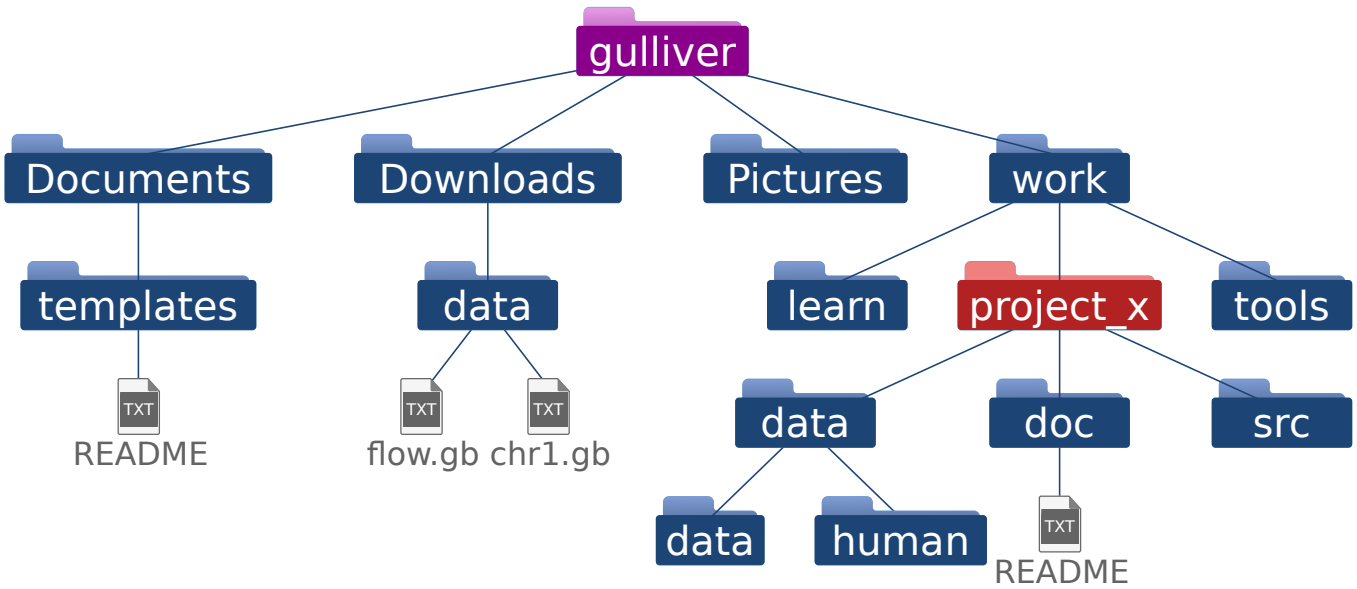
Copy files recursively

```
$ cp -r ../../Downloads/data data
```



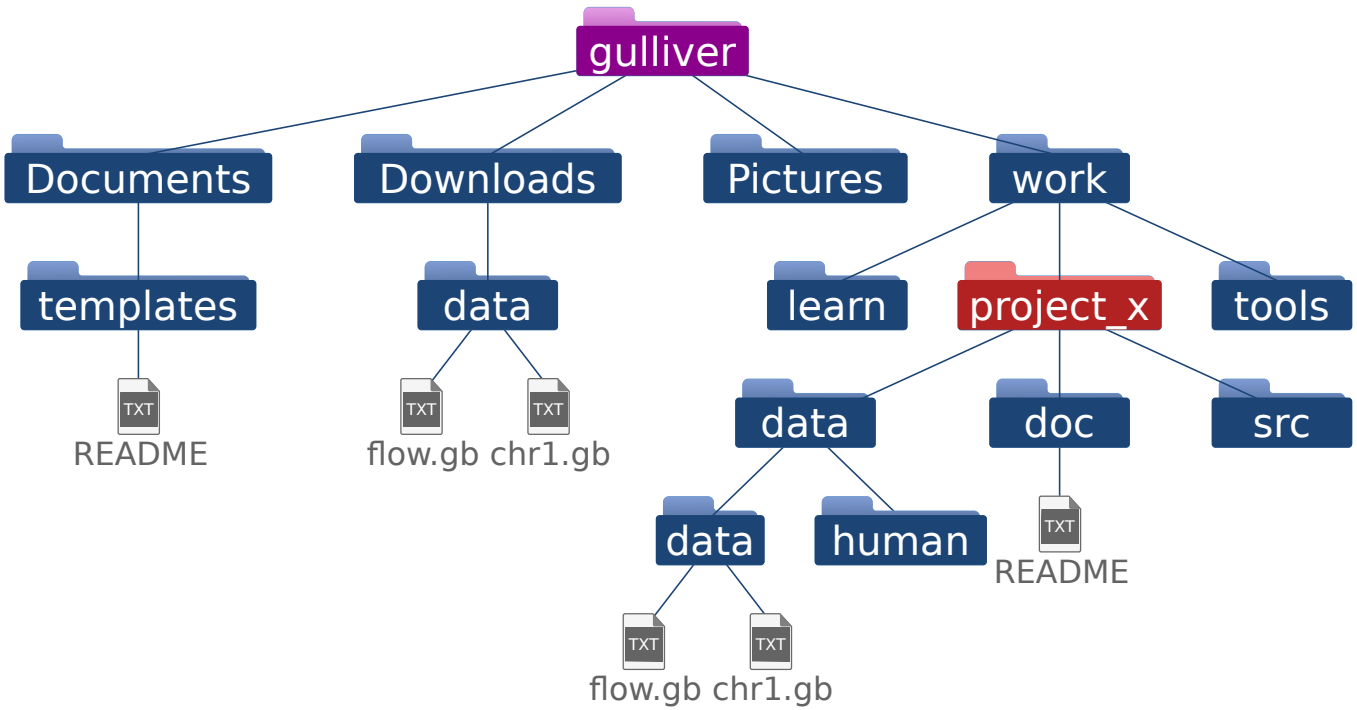
Copy files recursively

```
$ cp -r ../../Downloads/data data
$ ls data
data human
$
```



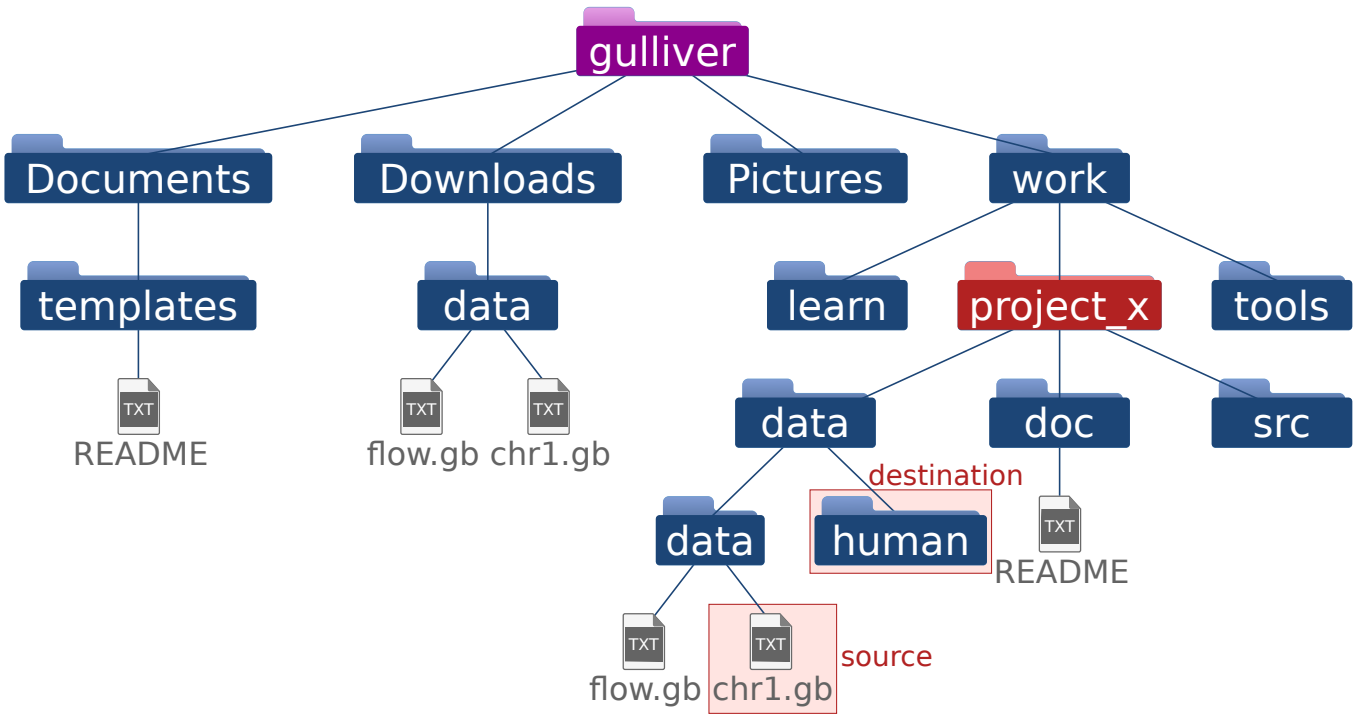
Copy files recursively

```
$ ls data/data
flow.gb chr1.gb
$
```



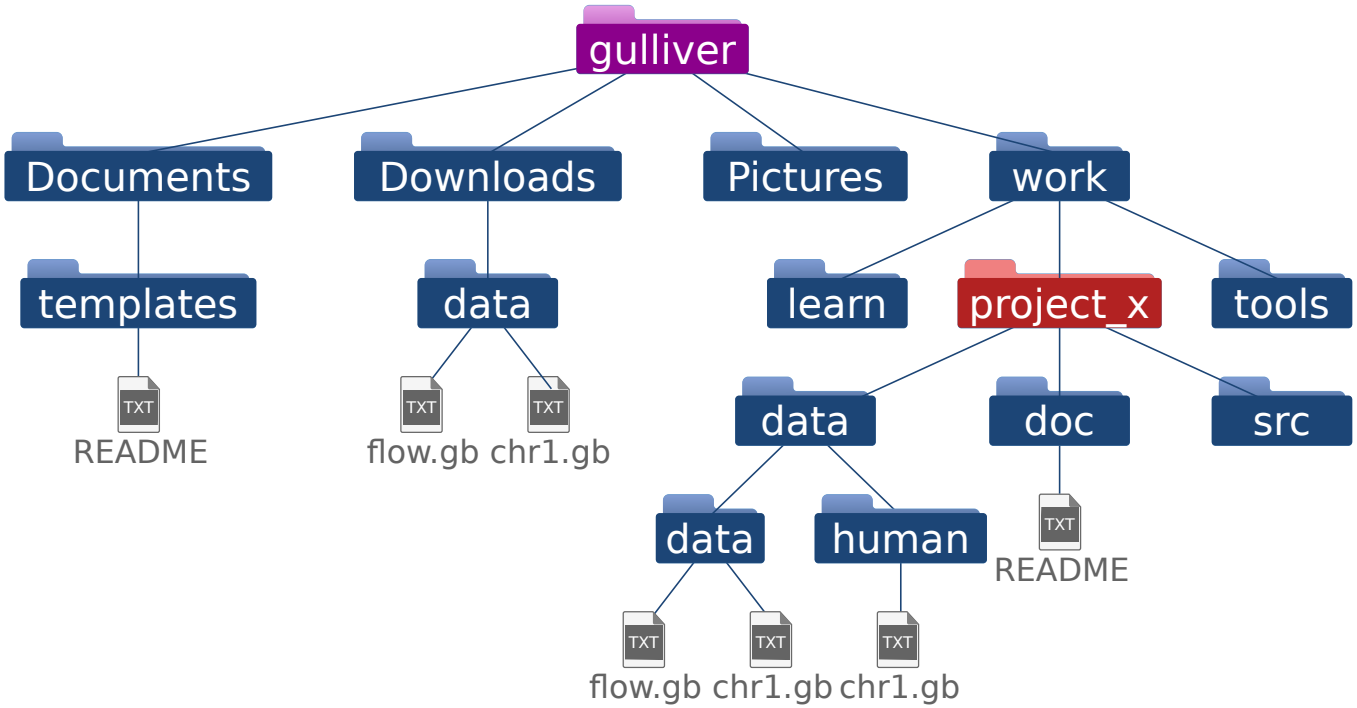
Move files

```
$ mv data/data/chr1.gb data/human
```



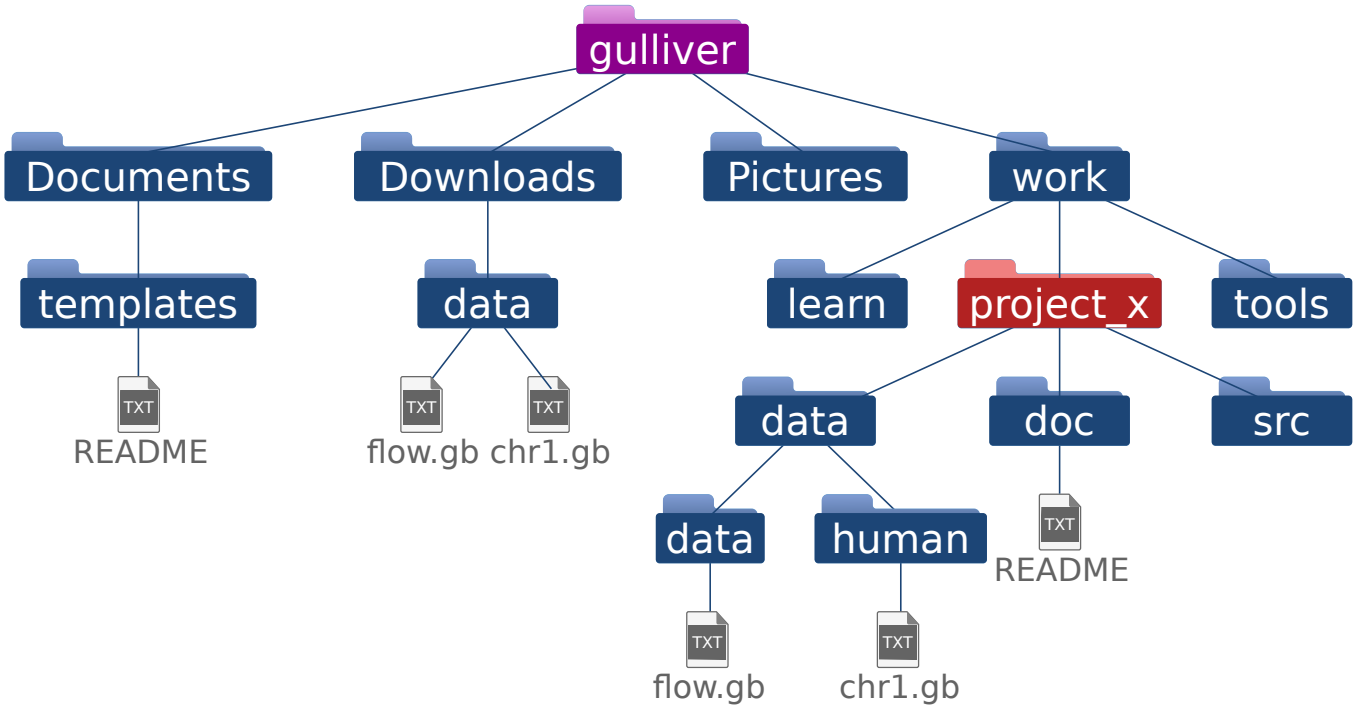
Move files

```
$ mv data/data/chr1.gb data/human
$ ls data/human
chr1.gb
$
```



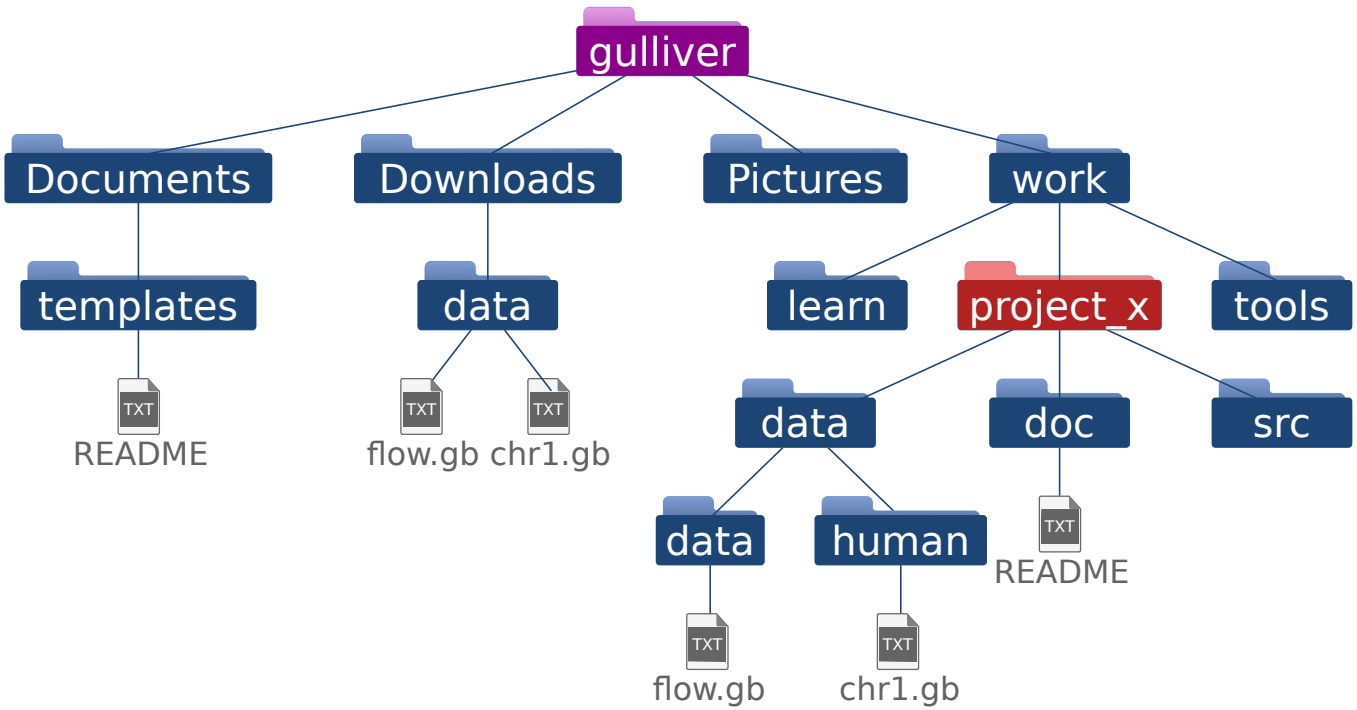
Move files

```
$ ls data/data
flow.gb
$
```



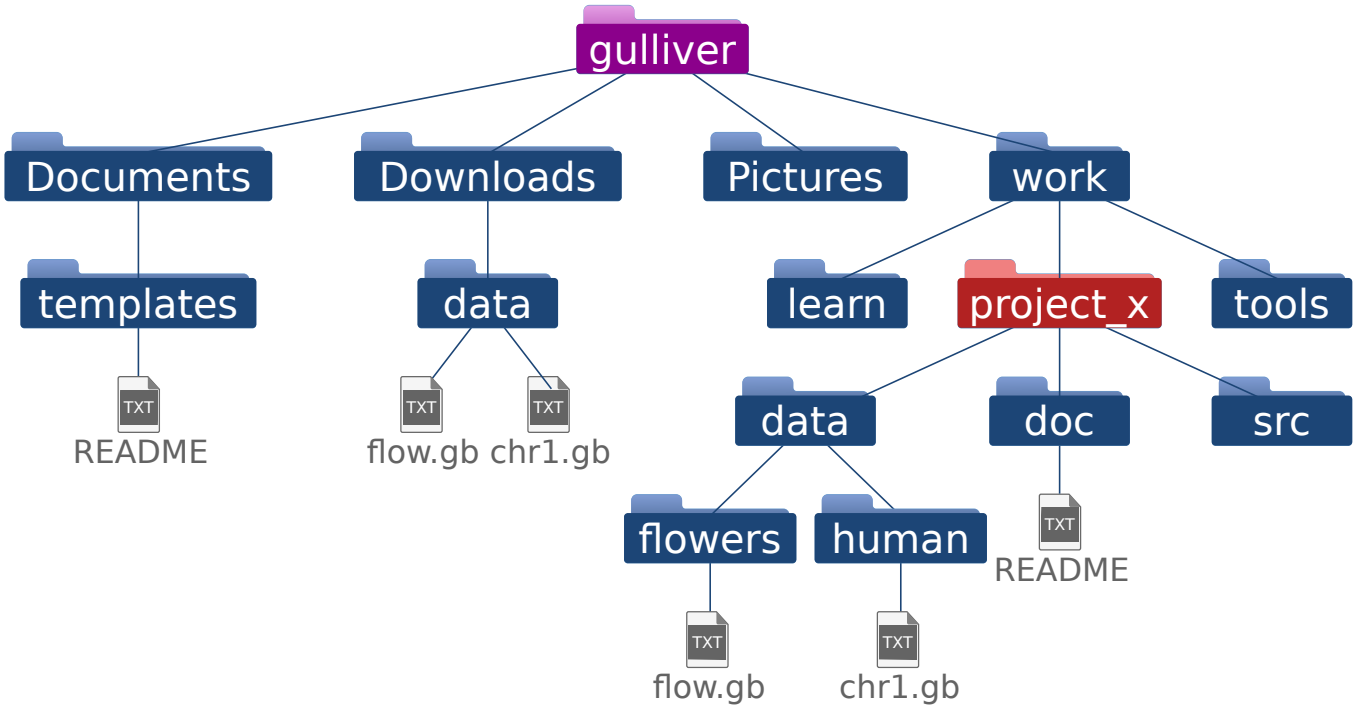
Rename files

```
$ mv data/data data/flowers
```



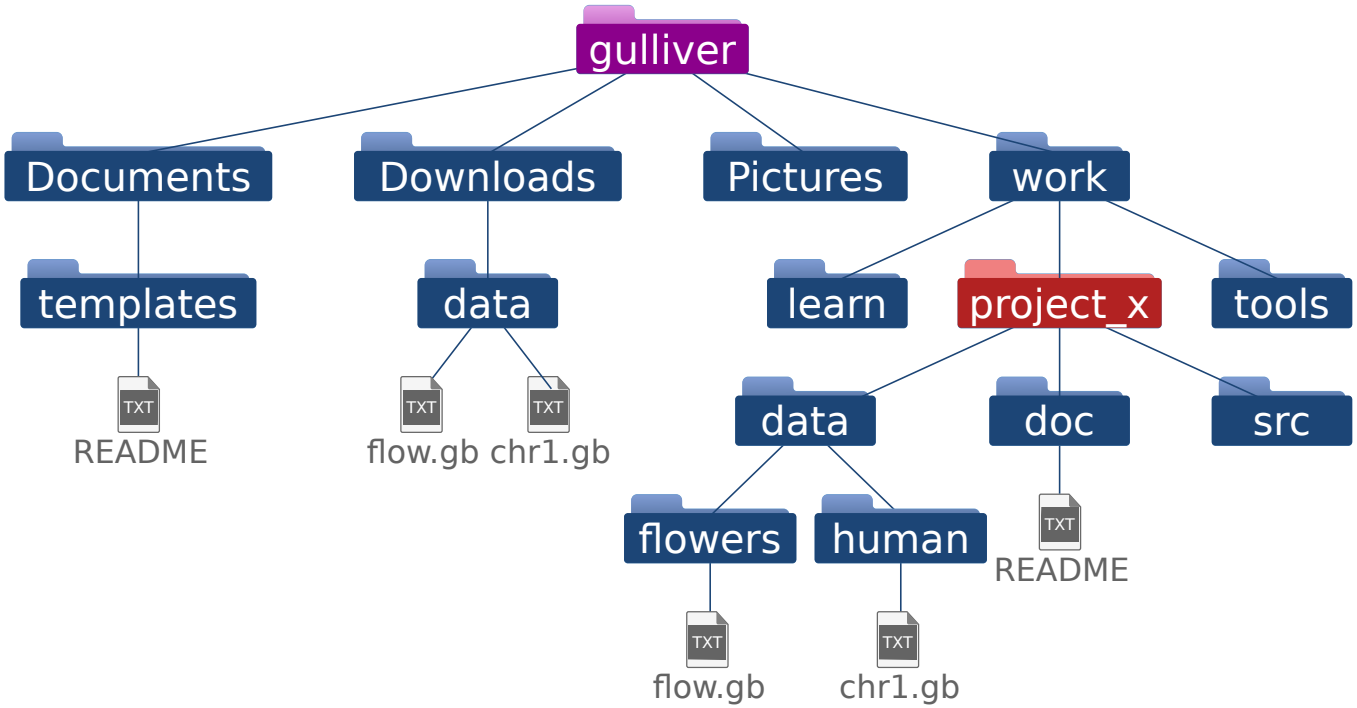
Rename files

```
$ mv data/data data/flowers
$ ls data
flowers human
$
```



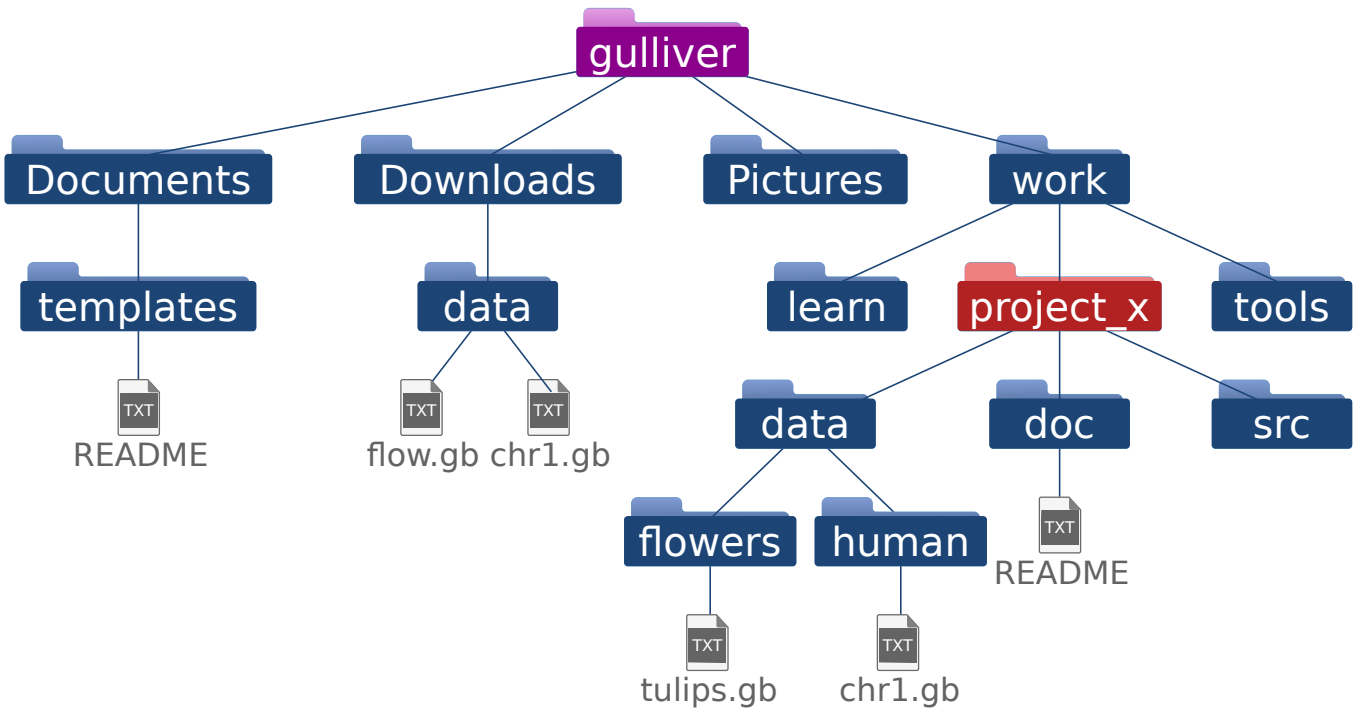
Rename files

```
$ mv data/flowers/flow.gb data/flowers/tulips.gb
```



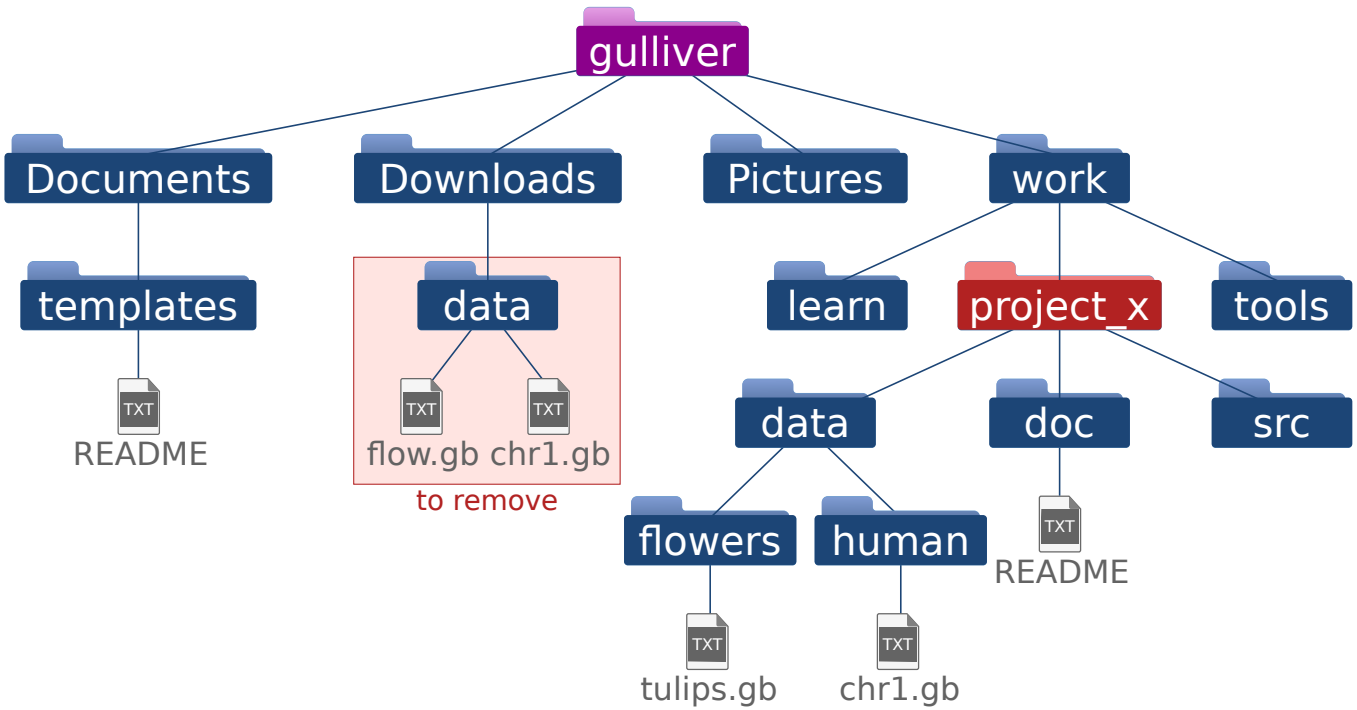
Rename files

```
$ mv data/flowers/flow.gb data/flowers/tulips.gb
$ ls data/flowers
tulips.gb
$
```



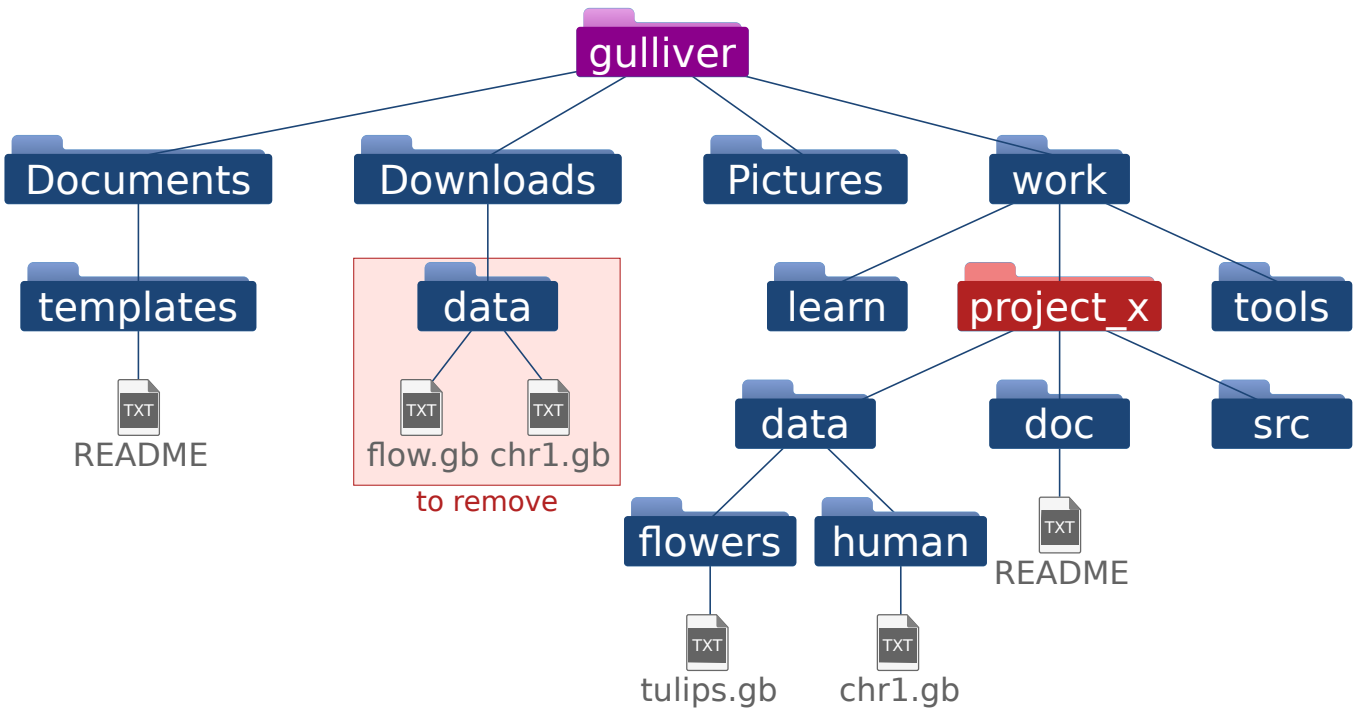
Remove files and directories

```
$ rm ../../Downloads/data
```



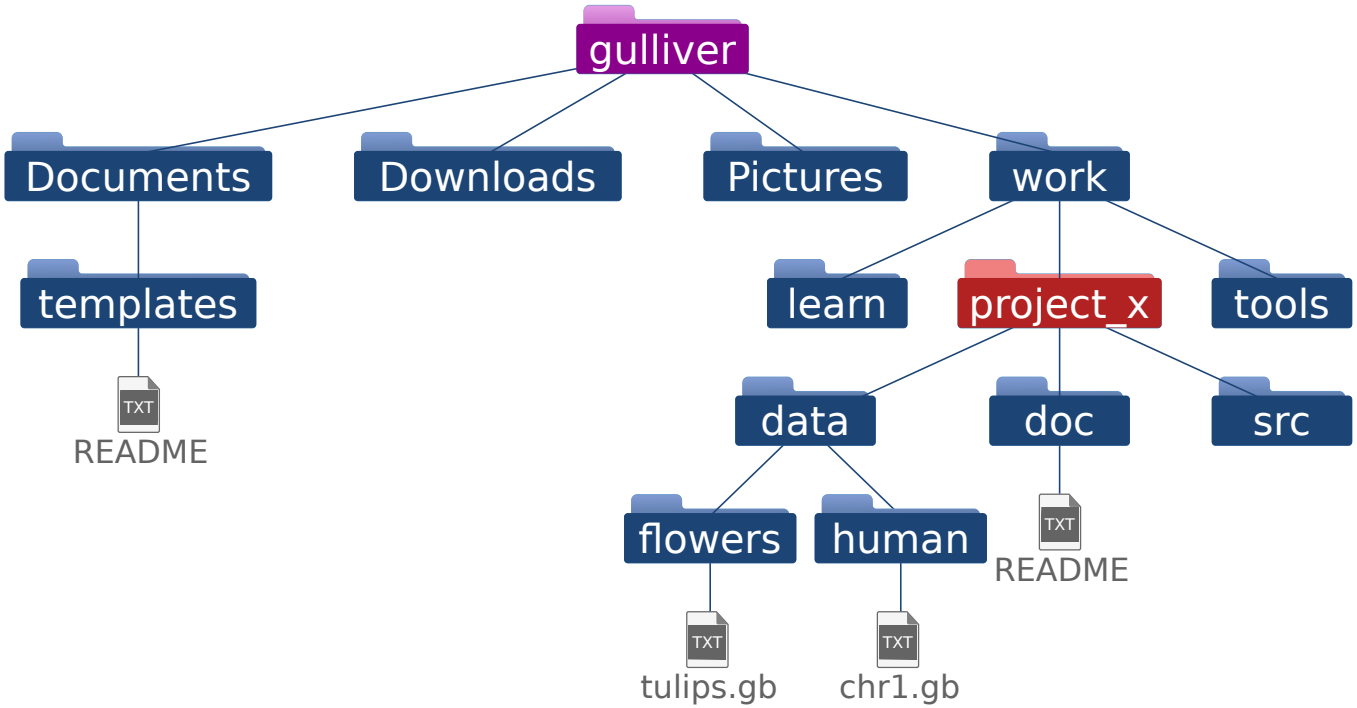
Remove files and directories

```
$ rm ../../Downloads/data
rm: cannot remove '../../Downloads/data': Is a directory
$
```



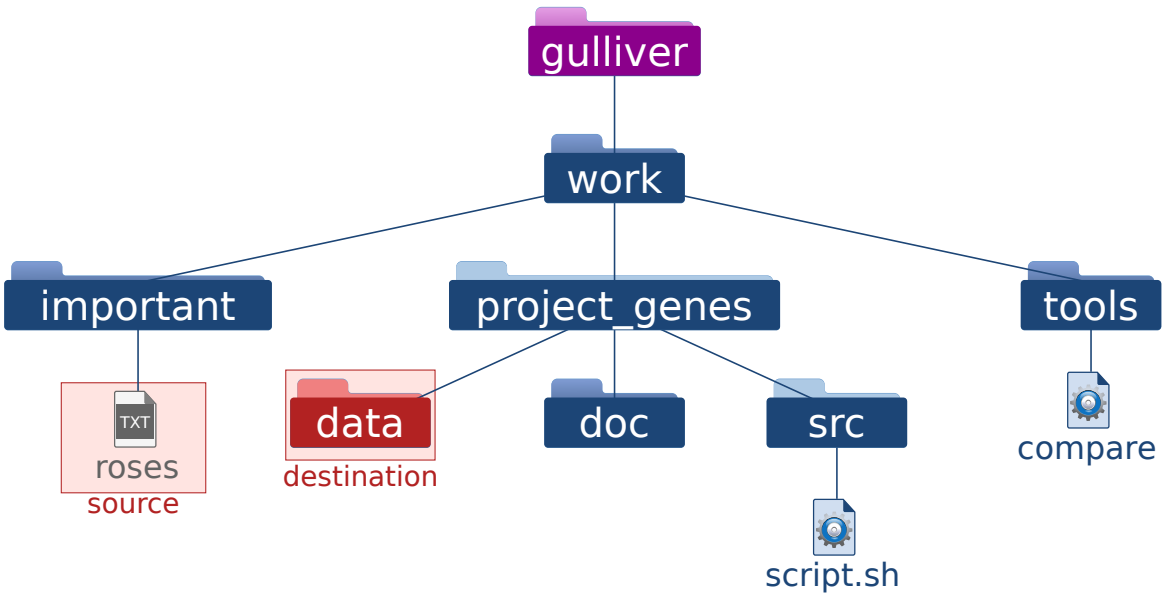
Remove files and directories

```
$ rm -r ../../Downloads/data
$ ls ../../Downloads
$
```



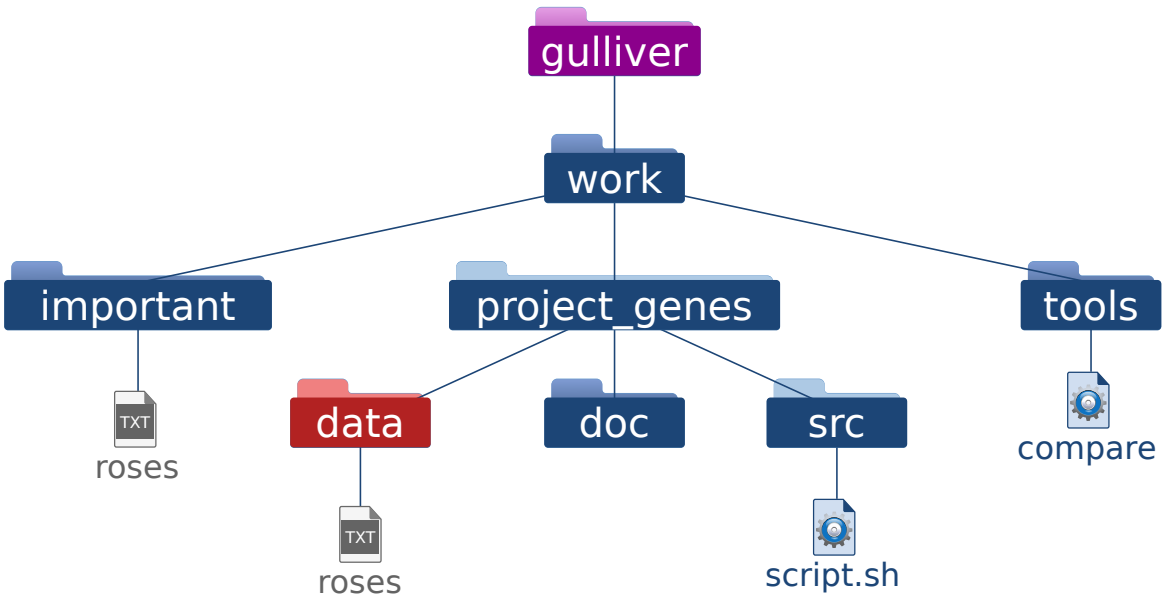
Refer to the current directory

```
$ cp ../../important/roses .
```



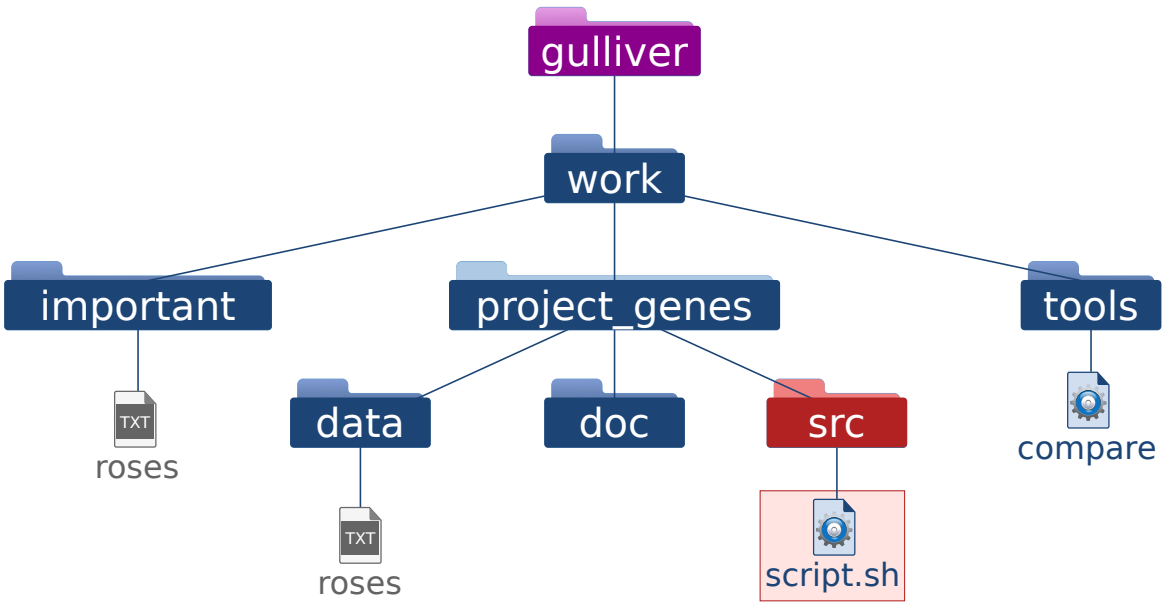
Refer to the current directory

```
$ cp ../../important/roses .
$ ls
roses
$
```



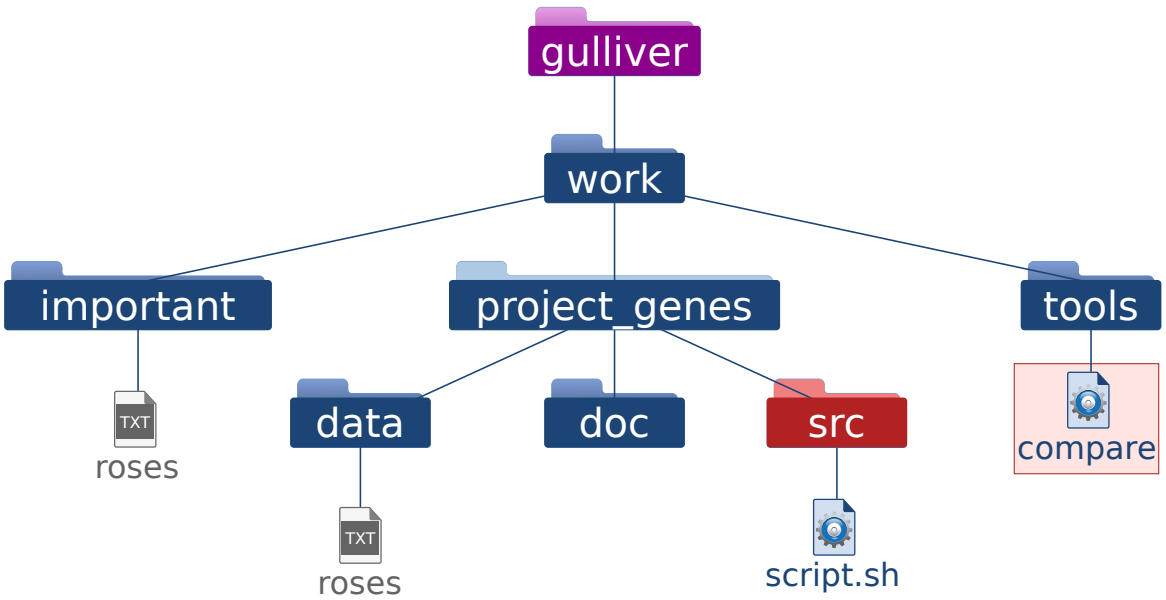
Run an application

```
$ ./script.sh
Hello gulliver! Job done!
$
```



Run an application

```
$ ../../tools/compare
Comparison perfomed: X is different than Y.
$
```



Quiz 1

What is the output of the closing `ls` command from below?

```
$ pwd
/home/zorro/data
$ ls
samples.dat
$ mkdir processed
$ mv samples.dat processed
$ cp processed/samples.dat ../samples-saved.dat
$ ls
```

1. samples-saved.dat processed
2. samples.dat processed
3. processed
4. samples-saved.dat

Quiz 1

What is the output of the closing `ls` command from below?

```
$ pwd
/home/zorro/data
$ ls
samples.dat
$ mkdir processed
$ mv samples.dat processed
$ cp processed/samples.dat ../samples-saved.dat
$ ls
```

1. samples-saved.dat processed
2. samples.dat processed
3. processed
4. samples-saved.dat

Inspect Files

`cat [option]... [file]...`

View the entire content of the input `files`.

`less [file]...`

View the `file` with scrolling allowed.

`head [option]... [file]...`

Print the first 10 lines of each `file`.

`tail [option]... [file]...`

Print the last 10 lines of each `file`.

cat

`cat [option]... [file]...`

View the entire content of the input `files`.

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
...
```

less

```
less [file]...
```

View the `file` with scrolling allowed.
Press `h` to get help, `q` to exit, and `/` followed by text to search.

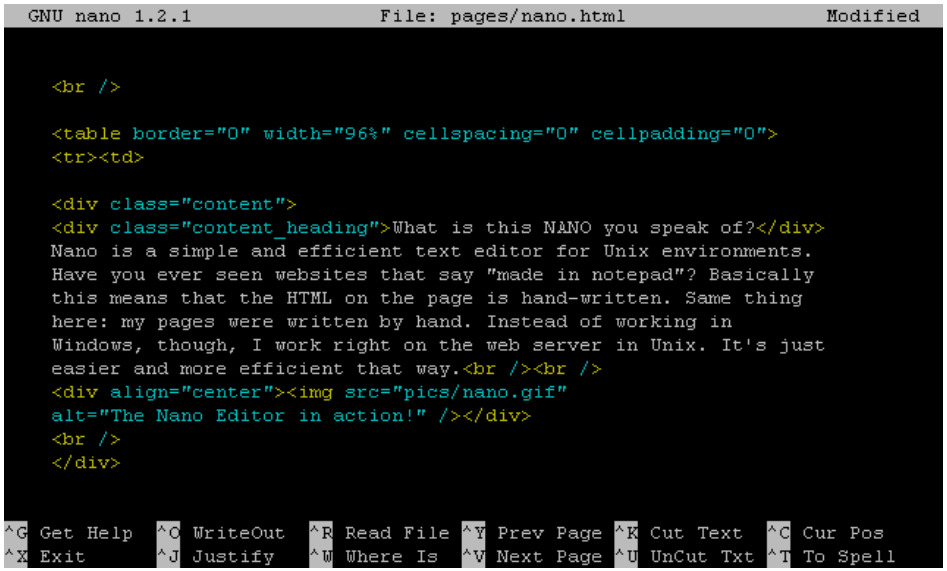
```
$ less /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
...
```

nano

A simple and easy text editor.
Open file to edit with:

```
nano file
```



The screen consists of:

- A header at the top.
- The text of the file being edited in the middle.
- A commands menu at the bottom.

Note that all keystrokes enter text into the file being edited.
Exception:

- Control (`Ctrl`, shown as ^) and Meta (`Alt` or `Cmd`, shown as `M-`) sequences.

nano

Highlithing, copying, cutting, and pasting:

- `Ctrl-^`: enter/exit highlight mode; use arrow keys next.
- `Meta-^`: copy (once highlighted).
- `Ctrl-k`: cut (once highlighted or cuts entire line if not).
- `Ctrl-u`: paste.

Search, undo, and redo:

- `Ctrl-w`: search for keyword.
- `Meta-u`/`Meta-e`: undo / redo.

Help, save, and exit:

- `Ctrl-g`: nano help; press `Ctrl-x` to exit the help mode.
- `Ctrl-o`: save file (write out).
- `Ctrl-x`: exit nano.

Disk Usage

du

`du [option]... [file/directory]...`

Summarize disk usage of the set of `files/directories`.

Some options:

- `-c, --total`
  Produce a grand total.
- `-h, --human-readable`
  Print sizes in human readable format (e.g., 1K 234M 2G).

```
$ du -ch /etc/passwd
4,0K /etc/passwd
4,0K total
```

File name expansion

Wildcards are used to substitute characters:

- ? - represent any single character.
- * - represent zero or more characters.

File name expansion

Wildcards are used to substitute characters:

- ? - represent any single character.
- * - represent zero or more characters.

```
$ ls
DC_000100.jpg   Documents       Downloads       flower.jpg
humans          genes           Music           programs
...
projects        Templates       Videos          Zagreb.jpg
$ ls *.jpg
DC_000100.jpg   flower.jpg      Paris.jpg       Zagreb.jpg
```

Escape wildcards with the backslash \.

Quiz 2

What is the output of the closing `ls` command from below?

```
$ ls -F
austria.gif  france.jpg   italy.gif   italy.jpg1   spain.jpg2
belgium.gif  gif/         italy.jpg   spain.gif
$ mv *.gif gif
$ rm *.jpg?
$ ls
```

1. The same as the first ls.
2. `austria.gif  gif  italy.gif  spain.gif`
3. `france.jpg  gif  italy.jpg`
4. `gif  italy.jpg1  spain.jpg2`

Quiz 2

What is the output of the closing `ls` command from below?

```
$ ls -F
austria.gif  france.jpg   italy.gif   italy.jpg1   spain.jpg2
belgium.gif  gif/         italy.jpg   spain.gif
$ mv *.gif gif
$ rm *.jpg?
$ ls
```

1. The same as the first ls.
2. `austria.gif  gif  italy.gif  spain.gif`
3. `france.jpg  gif  italy.jpg`
4. `gif  italy.jpg1  spain.jpg2`

locate

`locate [option]... [pattern]`

Outputs every path that contains the provided `pattern`.

```
$ locate zip
/bin/bunzip2
/usr/bin/zip
/var/lib/dpkg/info/bzip2.list
...
$ locate /usr/*zip
/usr/bin/zip
/usr/bin/gpg-zip
/usr/lib/klibc/bin/gzip
...
```

find

```
find [path]... [option]... [expression]
```

Search for files and directories based on the given `expression`.
Many options and ways to specify `expression` - check man page.

Example:
Search for all files ending with '.fasta' in directory 'mascot' that where last modified more than 2 years ago:

```
$ find /nfs/mascot -type f -name *.fasta -mtime +730
/nfs/mascot/sequence/Cdifile630/old/25374.C_difficile_6.fasta
/nfs/mascot/sequence/S_GP_0808/current/GeneDB_Smansoni_P.fasta
/nfs/mascot/sequence/S_GP_4.0h/old/GDB_Smansoni_P.v4.0h.fasta
```

grep

```
grep [option]... [pattern] [file/directory]...
```

Search text `files` for the `pattern` occurrence and output any line containing a match.

Some options:

- `-r, --recursive`
  Search recursively in all files of the provided `directory`.
- `-i, --ignore-case`
  Do not distinguish between upper and lower case characters.
- `-v, --invert-match`
  Invert the sense of matching to select non-matching lines.
- `-c, --count`
  Suppress normal output. Instead print the number of matches.

grep

```
$ grep -r TAF data/fasta_files
./A0A0A0MTS7.fasta:LVISMTFADDAGTAFIVVRNKHGETSASASLLMKSQQEMLY
./A0A0A0MTS7.fasta:KIAWYKDGKRIKHGERYQTAGLQDGRASLREGIYTAFASNI
./A0A0A0MTS7.fasta:IVAGSHTAFIQWFKDDQEISASEKYKFSFHSQLEGTDSGTY
./A0A0A0MTS7.fasta:QIEVTSSFTMLVIDNVTRFDSGRYNLTLETAFVRVLDSPSA
./P20848.fasta:IIRVPMINHLGRFDIHRDRELSTAFLAQHYVGNAPDPKKMWQ
$ grep -rc TA data/fasta_files
data/fasta_files/A0A0A0MTS7.fasta:158
data/fasta_files/P20848.fasta:1
$ grep -cv TA data/fasta_files/A0A0A0MTS7.fasta
443
```

Search and Count

wc

`wc [option]... [file]...`

Print the number of lines, words, and characters for each `file`.

```
$ wc P20848.fasta A0A0A0MTS7.fasta
 8     16     533    P20848.fasta
 601   607    36662  A0A0A0MTS7.fasta
 609   623    37195  total
```

Output Redirection

`command > file`

Redirect the output of `command` to the `file`.

```
$ find /nfs/mascot -type f -name *.fasta -mtime +730 > fastas
$ cat fastas
/nfs/mascot/sequence/Cdifile630/old/25374.C_difficile_6.fasta
/nfs/mascot/sequence/S_GP_0808/current/GeneDB_Smansoni_P.fasta
/nfs/mascot/sequence/S_GP_4.0h/old/GDB_Smansoni_P.v4.0h.fasta
```

Note that the `>` will overwrite any existing `file`.

You can use `>>` to append, i.e., add to an existing `file`.

That's it!

Commands and options explained:

- `https://explainshell.com/`

Leiden University
Medical Center

Acknowledgements

Mihai Lefter
Jonathan Vis
Jeroen Laros

Practical session

Outcomes

- Manipulate, i.e., create, delete, move, and rename files and directories.
- Inspect and edit files content.
- Search for and within files.
- Run scripts.
- Redirect commands output.

Structure

- Multiple choice questions (max 10 min).
- Hands-on (30 min).