# Part III: operations on files, using wildcards and combining commands

# Wildcards

- Since the shell uses filenames so much, it provides special characters to help rapidly specifying groups of filenames

- A group of special characters are called wildcards allow selecting filenames based on pattern of characters

# Wildcards

| Wildcard | Meaning |
|---|---|
| * | Matches any characters |
| ? | Matches any single character |
| [!characters] | Matches any character that is not a member of the set characters |
| [characters] | Matches any character that is a member of the set *characters*. The set of characters may also be expressed as a *POSIX character class* such as one of the following:<br>**[:alnum:]** Alphanumeric characters<br>**[:alpha:]** Alphabetic characters<br>**[:digit:]** Numerals<br>**[:upper:]** Uppercase alphabetic characters<br>**[:lower:]** Lowercase alphabetic characters |

Source: http://linuxcommand.org

# Wildcards examples

| Wildcard | Meaning |
|---|---|
| a* | Any file name starting with a |
| * | All possible filenames |
| A*.fasta | All filenames that begin with A and end with .fasta |
| ????.vcf | Any filenames that contain exactly 4 characters and end with .vcf |
| [abc]* | Any filename that begins with "a" or "b" or "c" followed by any other characters |
| [[:upper:]]* | Any filename that begins with an uppercase letter. This is an example of a character class |

# find command

- The find command can be used to find files matching a given expression. It can be used to recursively search the directory tree for a specified name, seeking files and directories that match the given name.
- To find all files in the current directory and all its sub-directories that end with the suffix fa:
  - find . -name "*.fa"

Will display all .fa files in the current working directory

# Basics operation on files

- **sort**: reorder the content of a file “alphabetically”
  syntax: sort <filename>
- **uniq**: removes duplicated lines
  syntax: uniq <filename>
- **join**: compare the contents of 2 files, outputs the common entries
  syntax: join <filename1> <filename2>
- **diff**: compare the contents of 2 files, outputs the differences
  syntax: diff <filename1> <filename2>

# Sorting data

- **sort** outputs a sorted order of the file content based on a specified sort key (default: takes entire input)

  Syntax: sort <options> <filename>
- Default field separator: Blank
- Sorted files are used as an input for several other commands so sort is often used in combination to other commands
- For <options> see man

# Sorting data: examples

- Sort alphabetically (default option): sort <filename>
- Sort numerically: sort **-n** <filename>
- Sort on a specific column (n°4): sort **–k 4** <filename>
- **...**

# uniq command

- **uniq** outputs a file with no duplicated lines
- Uniq requires a sorted file as an input

  Syntax: uniq <options> <sorted_filename>
- For <options> see man
- Useful option is **-c** to output each line with its number of repeats

# Join command

- **join** is used to compare 2 input files based on the entries in a common field (called “join field”) and outputs a merged file
- join requires sorted files as an input
- Lines with identitical “join field” will be present only once in the output

  join <options> <filename1> <filename2>

- For <options> see man

# diff command

- **diff** is used to compare 2 input files and displays the different entries
- Can be used to highlight differences between 2 versions of the same file
- Default output: common lines not showed, only different lines are indicated and shows what has been added (**a**), deleted (**d**) or changed (**c**)

  diff <options> <filename1> <filename2>
- For <options> see man

# Commands outputs

- By **default**, the **standard output** of any command will appear to the **terminal screen**.
- Redirection of the output result to a file is possible.
- This is particularly useful for big files
- Syntax: command options filename.**in** **>** filename.**out**

# Outputs redirection

- If the file exists, the result will be redirected to it

```
$ cat ghandi.txt
The difference between what we do
and what we are capable of doing
would suffice to solve
most of the world's problems
$ cut -d' '  -f2,3 ghandi.txt
difference between
what we
suffice to
of the

$ cut -d' '  -f2,3 ghandi.txt > ghandi.txt.out
$ cat ghandi.txt.out
difference between
what we
suffice to
of the
```

- If the file does not exist, it will be automatically created and the result redirected to it.

# Commands combination

- The **standard output** of any command will be **one unique output**
- As seen previously, this output can be **printed** in the screen or **redirected to a file**
- However, the **output** result of a command can also be **redirected to another command**
- This is particularly useful when several operations are needed for a file, with no need to store the intermediate outputs

# Commands combination: example

- Combining several commands is done thanks to the use of a "|" character

- Structure:

**command1 options1 filename1.in |command2 options2 > filename.out**

- This can be done for as many commands as needed

# Download files from the web

- **wget** stands for "web get". It is a command line utility which downloads files over a netwrok
- It supports HTTP, HTTPS, and FTP protocols

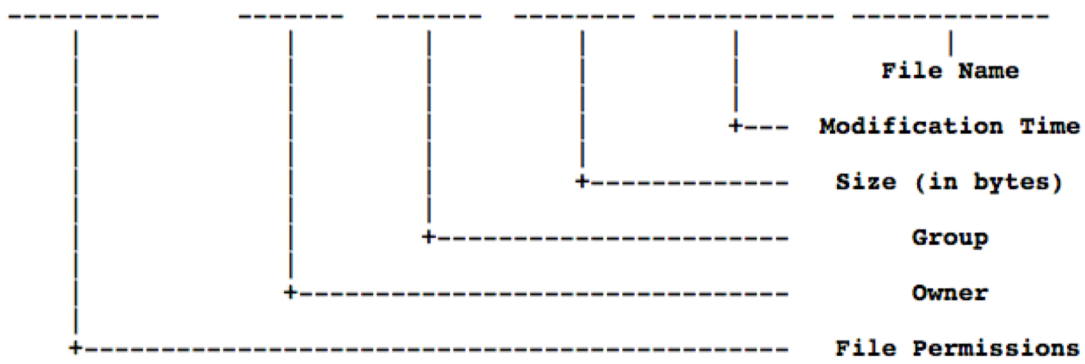
Syntax: wget [–options] [URL]

Let's try it:

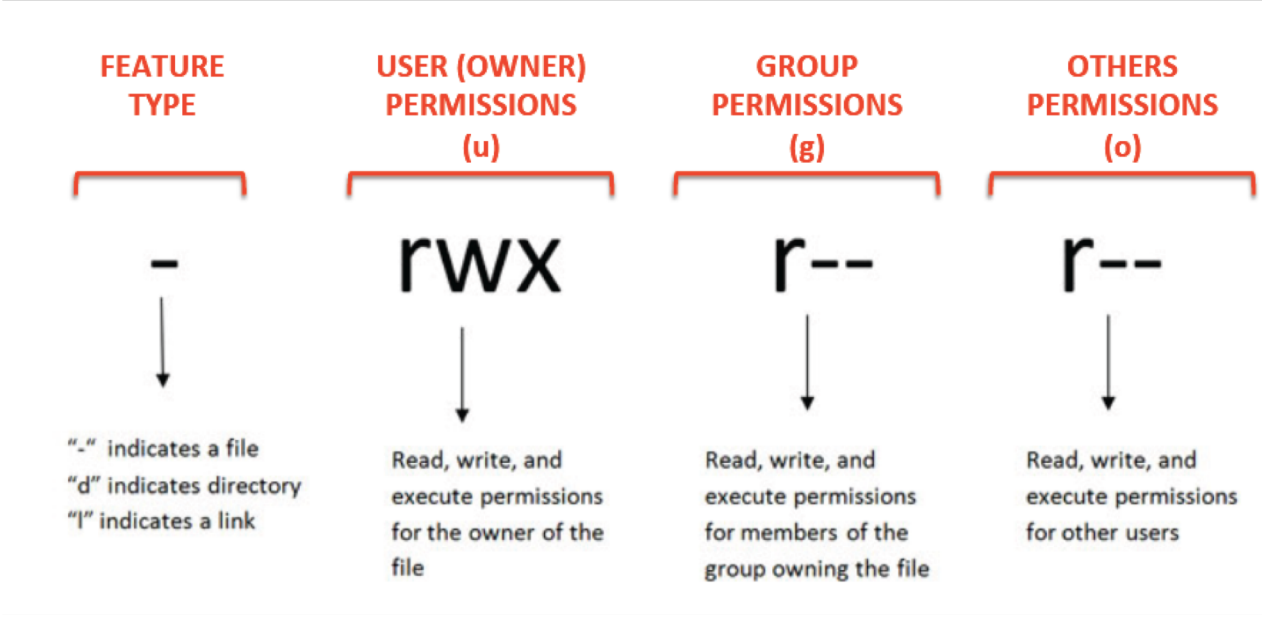
- Move to the directory Genomics and get the fasta file of P. falciparum from PlasmoDB
- Command: wget http://plasmodb.org/common/downloads/release-9.0/Pfalciparum/fasta/PlasmoDB-9.0_Pfalciparum_BarcodeIsolates.fasta

# Remember the ls -l example

```
drwxr-xr-x  2 amel  staff  68  7 aoû 18:15 Session1
drwxr-xr-x  2 amel  staff  68  7 aoû 18:16 Session2
-rw-r--r--  1 amel  staff  87  7 aoû 18:17 readme.txt
```



# Permissions are broken into 4 sections




Source: www.pluralsight.com

# Access permissions on files

- r indicates read permission: the permission to read and and copy the file
- w indicates write permission: the permission to change a file
- x indicates execution permission: the permission to execute a file, where appropriate

# Access permissions on directories

- r indicates the permissions to list files in the directory
- w indicates that users may delete files from the directory or move files into it
- x  indicates means the right to access files in the directory. This implies that you may read files in the directory provided you have read permission on the individual files

# chmod command

- Used to change the permissions of a file or a directory.
- Syntax: chmod  options permissions filename
- Only the owner of the file can use chmod to change the permissions
- Permissions define permissions for the owner, the group of users and anyone else (others)
- There are two ways to specify the permissions:
  - ✓ Symbols: alphanumeric characters
  - ✓ Octals: digits (0 to 7)

# Few tips

- Use tab completion - it will save you time!
- Build commands slowly!
- man the_name_of_a_command often gives you help
- Always have a quick look at files with less or head to double check their format
- Watch out for data in headers and that you don't accidentally grep some if you don't want them
- Regular expressions are wierd, build them up slowly bit by bit
- If you did something smart but can't remember what it was, try typing history
- Google is normally better at giving examples (prioritise stackoverflow.com results, they're normally good)

# Assignment 1