



Corso di Laurea di Primo Livello  
Scuola Universitaria Interfacoltà  
in Biotecnologie  
Università degli Studi di Torino



## Corso di Abilità Informatiche Secondo Modulo

AA 2008/2009

LABORATORIO INFORMATICO, LEZIONE 4:

METACARATTERI, FILE E PERMESSI, ALCUNI TOOL IMPORTANTI

### PREAMBOLO

**Importante:** Se non ricordate bene il funzionamento di un comando potete sempre consultare la sua pagina man. Se non ricordate il nome di un comando potete provare a cercarlo con apropos (ovvero `man -k`).

### ESERCIZIO 1: Metacaratteri

Visto l'importanza dell'argomento, riprendiamo gli esercizi sui metacaratteri.

1. Scaricate l'archivio *zip* che contiene alcuni file necessari per gli esercizi. Salvate il file nella home directory dell'utente. Aprite una shell e scompattatelo con il comando  

```
unzip <nomedelfilezip>
```
2. Cancellate il file *zip*. Scompattandolo è stata creata una cartella di nome `abInfo2_lab04`. In seguito tutti i file vengono creati all'interno di questa cartella che **va cancellata alla fine del laboratorio**.
3. Entrate nella cartella `abInfo2_lab04` e provate a farvi elencare tutti i file che stanno nella sottocartella `sequences`. Usando i metacaratteri ed il comando `ls`, provate a farvi elencare
  - a) i file che contengono sequenze nucleotidiche
  - b) i file che contengono sequenze di 5'UTR
  - c) i file che contengono sequenze di 5'UTR o di 3'UTR  
(spesso esiste più di un modo per farli elencare, quanti vi vengono in mente in questo caso?)
  - d) i file che contengono sequenze relative ai geni coinvolti nel *breast cancer* (BRCA)
  - e) i file che contengono trascritti di geni con nomi che iniziano per B o V
  - f) i file che contengono sequenze proteiche di proteine con nomi che NON iniziano per B
  - g) i file che contengono esoni o 3'UTR.

## ESERCIZIO 2: Tool utili – wc, file, diff, head, tail

In questo esercizio conoscerete alcuni comandi utili che non avete ancora visto a lezione. Date un'occhiata alle pagine *man* dei singoli comandi per sapere di più del loro funzionamento.

1. `wc` (da “word count”) è un comando che vi permette di contare il numero di righe, parole e/o caratteri in un file di testo.
  - a) Usate il comando (una volta sola) per scoprire il numero totale di righe, parole e caratteri contenuti negli **abstract** che si trovano nella cartella `test-files`.
  - b) Quale è l'abstract che contiene la riga più lunga (scoprite con `man wc` quale opzione dovete usare per farvi restituire la lunghezza massima delle righe di un file).
  - c) Usate il comando in modo da farvi restituire il solo numero di righe del file `write_lines.sh`. Il comando ha contato anche le righe vuote o sono state ignorate?
2. Il comando `file` vi permette di scoprire di che tipo sia un file. Provate il comando sui seguenti file:
  - a) `/home`
  - b) `/bin/ls`
  - c) `logoCBUatMBC.jpg`, `manpage_diff.txt` e `write_lines.sh` (nella cartella `test-files`)
  - d) `BRCA1_nuc_transcripts.fasta` (nella cartella `sequences`). Notate il risultato diverso da quello ottenuto per `manpage_diff.txt`, nonostante il fatto che entrambi siano dei file di testo.
3. Per verificare ed individuare le differenze tra due (o più) file potete usare il comando `diff` nel seguente modo:

```
diff <file1> <file2>
```

Guardate le differenze tra il file `manpage_diff.txt` che trovate nella cartella `test-files` e i quattro file simili che contengono delle modifiche: `manpage_diff_modified-[1-4].txt`  
L'output del comando `diff` indica in quale riga sono state incontrate delle differenze e di che tipo sono (a=added, d=deleted, c=changed). Esempio: “144a145” indica che tra le righe 144 e 145 del primo file è stata aggiunta una riga nel secondo file. Provate a riconoscere tutte le modifiche che sono state fatte nei 4 file simili a `manpage_diff.txt`.
4. I comandi `head` (capo, testa) e `tail` (coda) servono per far stampare sullo schermo solo la prima o l'ultima parte di un file di testo. Come default vengono stampate le prime 10 o le ultime 10 righe. Quale parametro si può usare per cambiare il numero di righe da stampare? Provate a farvi restituire:
  - a) le prime 20 righe del file `manpage_diff.txt`
  - b) la prima riga del file `write_lines.sh`
  - c) le prime due righe del file `Pellegrino_abstract.txt`
  - d) la prima riga di ogni file che contiene un'abstract (scrivendo un solo comando)
  - e) l'ultima riga del file `Pellegrino_abstract.txt`. Siete sicuri che sia una sola riga?
5. Il comando `tail` può essere molto utile anche per controllare un file di testo mentre viene ancora scritto (alcuni programmi per esempio scrivono dei log file per tener traccia della loro esecuzione). Trovate la descrizione dell'opzione `-f` (follow) del comando `tail`. E provate il suo funzionamento nel seguente modo:
  - I) nella shell che state usando adesso (shell 1): entrate nella cartella `test-files`
  - II) aprite un nuovo terminale con una seconda shell (shell 2) e entrate nella stessa cartella
  - III) nella shell 1: a) lanciate lo script `write_lines.sh` digitando: `./write_lines.sh`
    - b) interrompete l'esecuzione dello script (premendo `Ctrl+C`)
    - c) lanciatelo di nuovo redirezionando però il suo output in un file:

```
./write_lines.sh > output.txt
```

(faremo più esercizi sulla redirectione nel prossimo laboratorio)

IV) nella shell 2: mentre lo script continua a scrivere delle righe nel file `output.txt` (nella shell 1), potete verificare il proseguire della scrittura dalla shell 2 usando il comando `tail -f` nel seguente modo:

```
tail -f output.txt
```

V) nella shell 1: interrompete di nuovo l'esecuzione dello script. Cosa succede con il `tail -f` nella seconda shell?

### ESERCIZIO 3: file, cartelle e permessi

Linux è un sistema multi-utente, cioè viene spesso usato da più di un utente, anche contemporaneamente. In un tale sistema è importante poter definire dei permessi di accesso a file e cartelle per i singoli utenti (ad esempio per proteggere dati privati o per non fare agli utenti normali cambiare tutte le configurazioni del sistema). In seguito facciamo qualche esercizio sull'argomento.

1. Usate l'opzione `-l` del comando `ls` per verificare i permessi dei file in `test-files`. Per cosa stanno `r`, `w` e `x`? Per quale motivo appaiano in tre gruppi, uno solo non basterebbe?
2. Trasformate i permessi dei file in `test-files` nella loro forma ottale (es: `777` corrisponde a `rw-rwxrwx`). Trasformate anche i seguenti permessi nelle loro forma ottale:

```
-rw-r--r--  
-rw-----  
-r--r-----  
-rwxr-xr--
```

Cosa significano questi permessi? Chi può fare cosa?

3. Usate il comando `chmod` (guardate la pagina *man*) sullo script `write_lines.sh` per *togliere* il permesso di esecuzione (usando la *forma simbolica* per specificare il permesso) del gruppo e degli altri utenti. Verificate quali siano adesso i permessi. Potete ancora eseguirlo? *Togliete* anche i permessi di esecuzione del proprietario (che siete voi). Potete ancora eseguirlo?
4. Usando la *forma simbolica* *settate* i seguenti permessi per lo script `write_lines.sh`:  
Per il proprietario/utente e per il gruppo: permessi di lettura e di esecuzione; per tutti gli altri utenti: permessi di sola lettura. *Settate* questi permessi usando `chmod` una sola volta. Verificate che i permessi siano settati in modo corretto. Poi *aggiungete* il permesso di scrittura solo per l'utente stesso. Verificate.
5. Usando la *forma ottale* per specificare i permessi:
  - a) ripristinate i permessi di esecuzione di `write_lines.sh` solo per il proprietario e il gruppo, ma non per gli altri utenti. Aggiungete anche il permesso di lettura per il gruppo. Fate il tutto scrivendo una sola riga di comando (cioè usando `chmod` una sola volta).
  - b) togliete il permesso di scrittura (di tutti gli utenti) del file `output.txt` creato in un esercizio precedente e poi provate a cancellarlo. Fate attenzione al messaggio di conferma, è diverso dal messaggio di conferma se cancellate un'altro file con il permesso di scrittura settato? Provate a cancellare il file `logoCBUatMBC.jpg` e confrontate il messaggio di conferma.
  - c) create un file vuoto (con `touch`) e togliete il permesso di scrittura per tutti gli utenti. Provate a redirectione l'output di `write_lines.sh` nel file appena creato (con `./write_lines.sh > ilvostrofilevuoto`). Basterà ripristinare il permesso di scrittura per il gruppo (a cui appartiene anche il proprietario, cioè voi) per farlo funzionare?
6. I proprietari dei file: negli esercizi seguenti avete modificato i permessi di file che sono di vostra proprietà. Provate a scoprire chi sono i proprietari e i gruppi dei seguenti file o delle seguenti

cartelle:

```
/bin/ls
/etc/cups/*
/home/
```

Provate a togliere il permesso di esecuzione per il gruppo e gli altri utenti del file `/bin/ls`. Che risultato vi aspettate e perché? Chi ha il diritto di cambiare i permessi di un file?

7. Create una cartella di nome `sequences_copy` all'interno di `test-files`. Copiate tutte le sequenze dalla cartella `sequences` nella cartella appena creata (con un singolo comando). Verificate quali permessi siano settati per la cartella creata e per i file copiati. Dopodiché togliete il permesso di lettura del gruppo e degli altri utenti alla cartella e in modo ricorsivo (opzione `-R`) anche per le copie delle sequenze (usando un singolo comando). Verificate il risultato.
8. Permessi di una cartella: togliete il permesso di lettura della cartella `sequences_copy` anche per l'utente. Provate a farvi listare il suo contenuto senza spostarvi in essa con `cd`. Funziona? Spostatevi nella cartella e provate a listare il suo contenuto. Funziona? Tornate nella cartella `test-files` e risettate i permessi di lettura per `sequences_copy` e tutti i file che contiene almeno per l'utente stesso. Verificate con `ls` che funzioni. Adesso togliete invece il permesso di esecuzione per l'utente. Fate di nuovo `ls` sulla cartella per farvi listare il suo contenuto. Poi provate a leggere uno dei file che contiene usando il comando `less` (potete anche usare `cat` o `more`). Poi provate ad entrare nella cartella con `cd`. Risettate il permesso di esecuzione sulla cartella e provate di nuovo ad aprire uno dei file che contiene con `less`.

Alla fine di tutti gli esercizi, ricordatevi di cancellare la cartella `abInfo2_lab04` con tutti il suo contenuto (possibilmente con un singolo comando).