

Bioinformatica II
LM Biologia Evoluzionistica, Università di Padova
Docenti: Dr. Giorgio Valle, Dr. Stefania Bortoluzzi
Esercitazione 1

Padova, 11 novembre 2014

GUIDA

SISTEMA OPERATIVO: LA PIATTAFORMA LINUX

Attualmente, la maggior parte dei computer monta di default un sistema operativo Windows in una delle sue numerose versioni o aggiornamenti.

Nonostante la diffusione dei sistemi Microsoft, il nuovo “appeal Apple” acquisito con iPod, iPhone, iMac ne ha determinato una maggior popolarità a discapito dei primi.

La ricerca si sa...preferisce investire in apparecchiature ed esperimenti. E' bene sapere che esistono sistemi operativi “Open Source” (trad: gratis) in grado di soddisfare le stesse necessità operative e che vengono sviluppati con interfacce che sempre più si avvicinano a quelle dei personal computer maggiormente utilizzati. Sono le numerose distribuzioni basate su sistemi operativi Linux.

Nonostante al primo impatto possano sembrare ostici (“robe da nerd”, forse), sono sufficienti un po' di pratica e pazienza per destreggiarsi tra le loro funzioni in modo del tutto simile agli altri.

La domanda sorge spontanea: perché ricorrere ad un sistema Linux piuttosto che Windows ? (chi usa Mac...usa Mac).

• **PRO:**

- sono migliaia e con le caratteristiche più svariate.
- vengono continuamente sviluppati. Sarà possibile avere sempre versioni aggiornate.
- **SONO GRATUITI:** è possibile scaricarne una versione e lanciarla sul proprio computer (vedi oltre).
- Dispongono di numerosi applicativi (compresa la versione gratuita di Word/Excell/PowerPoint) che non fanno rimpiangere gli eseguibili Windows.
- La maggior parte dei programmi ed algoritmi per la biologia non hanno interfacce grafiche (funzionano da riga di comando). Linux e il suo terminale rappresentano forse la via più semplice (e talvolta, unica) per poter utilizzare uno di questi software

- **CONTRO:**

- richiedono quel minimo di pazienza che sicuramente tutti noi abbiamo già avuto affrontando il nostro primo computer Windows
- richiedono un minimo di competenza informatica per l'installazione/gestione dei programmi (spesso mancano di interfaccia grafica).

LANCIARE UN O.S. LINUX

Un sistema operativo Open-Source Linux-Based può essere lanciato ed eseguito su ogni computer. Generalmente, MacBook-users non lo richiedono in quanto spesso (non sempre!) le stesse funzioni possono essere svolte analogamente ad un sistema Linux.

E' possibile scaricare uno degli innumerevoli sistemi di cui ne vengono riportati alcuni:

Scaricata la versione **.iso**, è possibile utilizzare uno dei programmi per la creazione di un "bootable-disk". Programmi come **Unetbootin** (<http://unetbootin.sourceforge.net/>), o **Yumi** (<http://www.pendrivelinux.com/yumi-multiboot-usb-creator/>) servono per questo scopo e dispongono di "procedure di creazione" assolutamente assistite ed automatizzate.

Il "bootable-disk" è un dispositivo, una chiavetta USB o un CD/DVD sul quale verrà "scompattato" il file .iso e che potrà essere caricato direttamente all'avvio del computer.

Brevemente, una volta creato il DVD o la USB-stick:

- riavviare il proprio computer
- leggere ciò che riportano le prime schermate (e.g. "Press DEL to enter SETUP," oppure "press F2" o "F8")
- clickare ripetutamente il tasto indicato fino a quando si apre la pagina del BIOS (funzioni base del computer).
- con i tasti "freccia", muoversi fino alla finestra "Boot" e premere invio. A display, è evidenziata la lista di componenti che il vostro computer utilizzerà per caricare il sistema operativo, ordinate a seconda delle priorità.
- Cambiare le priorità a seconda si disponga di un DVD o di una chiavetta USB ponendole in cima alla lista.
- Salvare, confermare ed uscire (frequentemente con il tasto "F10")
- Il computer si riavvierà andando a cercare nel dispositivo che gli avete specificato il sistema operativo (Linux, nel nostro caso).

Il tutto può essere seguito passo passo su questo tutorial

<https://www.youtube.com/watch?v=smL4hDBrN2A>

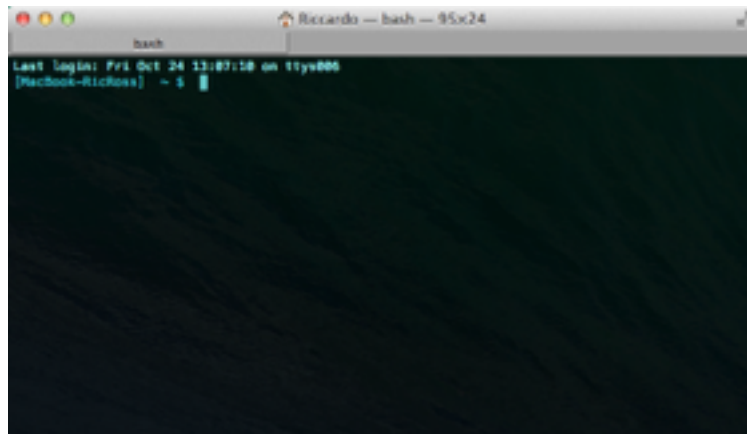
INTRODUZIONE ALLA SHELL

Sistemi operativi comunemente utilizzati (Windows, Mac), funzionano attraverso l'interazione visiva dell'utente con l'ambiente grafico.

Grazie al puntatore del mouse, è possibile muoversi liberamente all'interno del proprio computer inviando comandi sotto forma di "click" per aprire cartelle, files e lanciare programmi.

I "dati fisici" di movimento registrati dal mouse non sono altro che **stringhe di caratteri** che vengono interpretati dal sistema operativo e convertiti in "azioni visive".

Tuttavia esiste una via alternativa che permette di gestire il proprio computer senza avvalersi di mouse ed ambiente grafico. L'interfaccia di comunicazione è detta "**shell**" (**terminale**) e rappresenta la versione scritta di ciò che normalmente siamo abituati a vedere sotto forma di immagine (o icona).

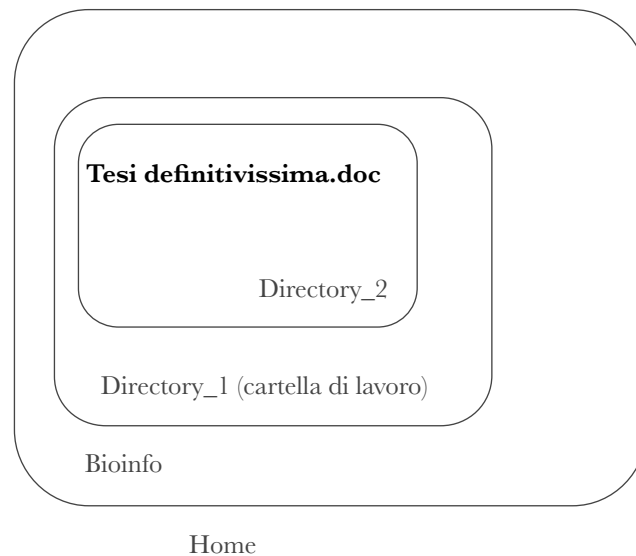
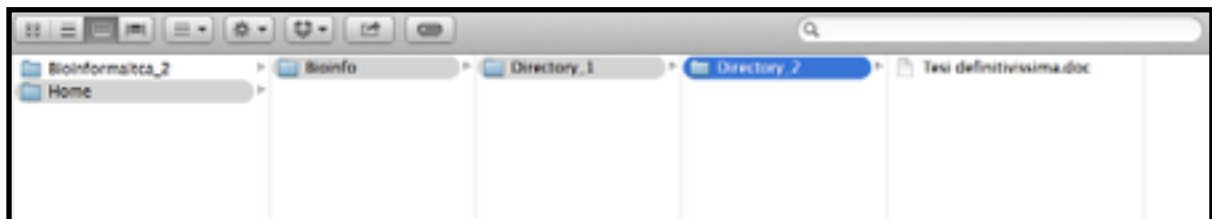


Esercizio: trovare ed aprire il terminale in un computer Linux.

Trovato ed aperto il terminale, potete vedere un percorso specificato in prossimità del cursore lampeggiante.

La directory (cartella) “Home” rappresenta il punto di partenza nonché la directory principale da cui è possibile navigare nel proprio sistema operativo. La “Home” viene specificata da un percorso **/Home/NomeUtente**.

Ciascuna cartella, nel linguaggio shell, viene racchiusa da due slash.



Come si “esprime” nel linguaggio Shell il percorso che dall’utente porta al file “Tesi definitivissima.doc”?

`/Home/Bioinfo/Directory_1/Directory_2/`

COMANDI PRINCIPALI

Come anticipato, il terminale rappresenta ciò che normalmente siamo abituati a vedere sotto forma di finestra e del suo contenuto in cartelle, files etc. Bastano pochi comandi per permetterne l'esplorazione.

Comandi principali:

comandi	significato
pwd	stampa il percorso ed indica dove l'utente si trova
ls	lista il contenuto della directory
ls -ltr	lista il contenuto della directory ordinandone il contenuto per data di creazione/
cd <percorso>	cambia directory (cartella di lavoro)
mkdir <nome_cartella>	crea una directory (cartella di lavoro) dove ci troviamo
rm <nome_file>	cancella un file
rm -r <nome directory>	cancella una directory (data "l'importanza" del livello, è necessario "forzarne" la rimozione con il comando -r)
less <nome_file>	visualizza un file di testo
nano <nome_file>	editor di testo
freccia su	clickando la "freccia su" appariranno in successione cronologica tutti i comandi impartiti
history	elenca tutti i comandi impartiti fino ad ora
Ctrl + lettera "C"	Killa un processo che sta correndo su una finestra
*	il carattere * indica un qualsiasi elemento presente nella directory e viene utilizzato in associazione con altri comandi
.	il "." indica la directory corrente e può essere utilizzato in combinazione con altri comandi

Dati questi comandi principali, è già possibile effettuare qualche operazione. **PROVATE!**

Esercizio: creare una serie di cartelle di lavoro

- aprire la finestra di esplorazione tramite ambiente grafico (il “Finder” per i Mac-Users, “Esplora Risorse” per utenti Windows).
- Aprire il terminale.
- Dimensionare le due finestre in modo che siano entrambe visibili sul monitor.
- Digitare in parallelo una serie di comandi da terminale (BLU) ed osservare il risultato grafico (ROSSO).

- Localizzare la propria posizione:

```
pwd
```

- Entrare nella cartella Home:

```
cd /Home/
```

- Doppio click nella cartella Home

Spiegazione: il comando cd (change directory) permette di accedere e spostarsi all'interno delle varie cartelle.

- Creare la cartella Directory_1

```
mkdir Directory_1
```

- Cosa succede nella versione grafica?

Spiegazione: il comando mkdir (make directory), permette di creare una cartella di lavoro.

- Entrare nella cartella appena creata

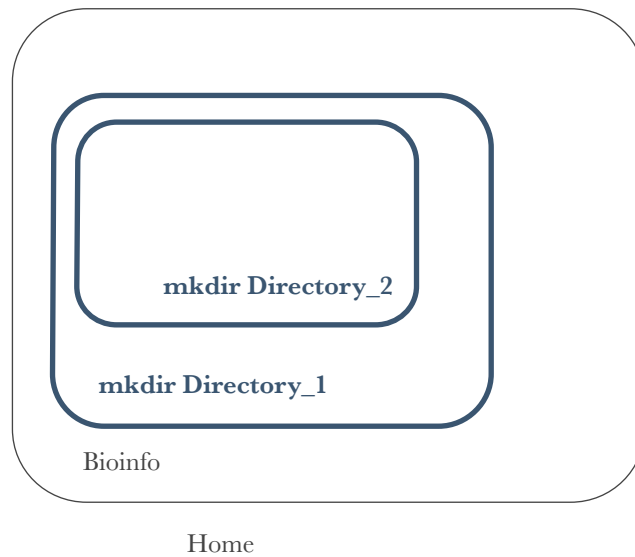
- Localizzare la propria posizione

- Doppio click nella cartella Directory_1

- Creare la cartella Directory_2

- Cosa succede nella versione grafica?

Se tutto è andato bene, sono state create due cartelle attraverso una successione di comandi:



Esercizio: creare un file di testo

- Nella `Directory_2`, aprire l'editor di testo

```
nano
```

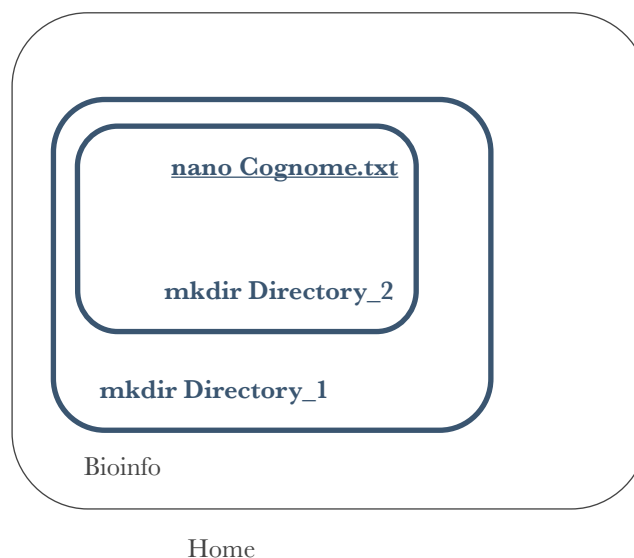
- digitare una riga di esempio (proprio nome e cognome)
- salvare digitando in successione `Ctrl` e il tasto `O` con il vostro cognome e l'estensione `.txt`
- uscire dall'editor digitando in successione `Ctrl` e il tasto `X`
- Assicurarsi che il file sia stato creato

```
ls
```

- Esplorare il file appena creato

```
less <nome file>
```

- Osservare se il file è stato creato utilizzando l'ambiente grafico



- Smanettare un po':
- Rimuovere il file appena prodotto

```
rm cognome.txt
```
- **Accertarsi che il file sia stato rimosso**
- Rifare il file

NOTA 1: è buona norma non utilizzare gli “spazi” per separare nomi composti di files ma utilizzare il simbolo “_”.

NOTA 2: la maggior parte dei files con cui vi confronterete, siano essi programmi che verranno eseguiti, esempi che si faranno o risultati di complessi calcoli, verranno rappresentati come **FILES TESTUALI**.

Un file testuale non è un documento Word. Si tratta di stringhe di caratteri visualizzati senza alcun tipo di formattazione. Questi files possono essere aperti con uno dei tanti “editor di testo” (“nano” ne è un esempio preinstallato su Ubuntu) e rappresentano il formato universale riconosciuto da tutti i computer e sistemi operativi.

Ad esempio, se vi è mai capitato di avere documenti Word che perdono immagini e formattazione passando da un computer ad un altro, da una versione di Windows ad un'altra, questo non può accadere con un file testuale. Ovviamente la grafica è scarna e meno appariscente rispetto al complesso mondo della “formattazione da tesi di laurea”.

SCORCIATOIE E TRUCCHI: LA SHELL PIU' VELOCE DEL MOUSE

Abbiamo visto come creare una cartella e abbiamo cominciato ad esplorare la shell. A questo punto, una serie di scorciatoie ci permetteranno di compiere azioni sui files, muoverci tra le directory e gestire il computer in modo estremamente veloce e pratico.

CAMMINARE TRA LE CARTELLE

Il comando “cd” permette di entrare in una cartella. Ma è possibile impartire un unico comando che ci permetta di andare da una posizione ad un'altra in un singolo click.

```
cd /Home/NomeUtente/Directory_1/Directory_2
```

La shell prevede l'utilizzo del tab per compilare in modo automatico il percorso senza dover scrivere interamente il nome della cartella. E' sufficiente “tabbare” per vedere completata la riga oppure basta specificare ulteriori parole nel caso siano presenti più cartelle o files che condividono le stesse lettere.

Ma se volessimo tornare indietro? La scorciatoia che nel terminale rappresenta una directory generica è rappresentata dal comando “../” (dot-dot slash) associato a “cd”.

Se dalla cartella Directory_2 volessimo tornare alla Home, possiamo digitare il percorso intero specificando ../ per ogni livello. Nell'esempio specifico:

```
cd ../
```

spiegazione: ci riporta dalla Directory_2 alla Directory_1, controllare la posizione

```
cd ../
```

spiegazione: ci riporta dalla Directory_1 alla cartella NomeUtente

```
cd ../
```

spiegazione: ci riporta dalla cartella NomeUtente alla cartella Home

oppure...

```
cd ../../../../
```

spiegazione: digitato in un'unica soluzione, ci riporta direttamente alla cartella Home.

TIPI DI FILES

Analogamente a Windows e Mac, anche un sistema operativo Linux dispone di diversi tipi di files a seconda di ciò che specificano. Le **estensioni** (.doc, .xls, .ppt, .jpeg etc.) ne consentono il riconoscimento automatico e l'associazione a programmi specifici consentendone l'esecuzione.

Linux (e la shell) sono caratterizzati da una "gestione elastica" dei files: ciascun documento, immagine, foglio di calcolo, può comunque essere esplorato (ma non eseguito).

Inoltre, a differenza di ciò che succede in un sistema Windows in mancanza di un software in grado di riconoscere ed eseguire una specifica estensione, Linux è molto meno restrittivo. Qualsiasi estensione può essere utilizzata (anche ".pippo", ".ciaociao", ".fasta") e spesso è possibile aprire un file od eseguire un comando indipendentemente dalla stessa.

OPERAZIONI SUI FILES

Come si procede normalmente, anche attraverso la shell è possibile rinominare, copiare, incollare e rimuovere files e cartelle.

Una serie di comandi permettono di operare sui files.

comandi	significato
less <nome_file>	visualizza un file di testo
mv <old_name> <new_name>	rinomina un file
mv <nome_file> <percorso>	sposta il file specificato in un'altra cartella
cp <nome_file> <new directory>	
nano <nome_file>	editor di testo
>	redirect: dirige il risultato di un comando su un file di testo
cat <file_1> <file_2> > <file_3>	concatena i due files (ponendoli uno di seguito all'altro) in un unico file
rm <nome_file>	rimuove un file ATTENZIONE, L'OPERAZIONE NON E' REVERSIBILE
wc -l <nome_file>	word-count ed opzione lines: riporta sulla shell il numero di righe che compone un file

Esercizio: azioni sui files

- Entrare nella cartella Home/NomeUtente
- Aprire l'editor di testo "nano"
- Digitare in colonna da 10 a 20 (a vostra discrezione) nomi di persona
- Salvare il file con un nome ed un'estensione a piacere, anche casuale

A questo punto, controllare se il file è stato salvato correttamente attraverso il comando "less". Nonostante nome ed estensione puramente casuali, il comando less permette la visualizzazione del contenuto del file. **RICORDATE L'UTILIZZO DEL TAB PER IL COMPLETAMENTO AUTOMATICO DEL PERCORSO!**

- Copiare il file appena creato nella cartella Directory_2

```
cp newFile /Home/NomeUtente/Directory_1/Directory_2
```
- Accertarsi che il file non sia più presente nella cartella. E' presente? Ipotesi?

```
ls
```
- Muoversi nella Directory_2 ed accertarsi che il file sia stato correttamente spostato

Ora nella Directory_2 dovrebbero essere presenti 2 files. Controllare se sono entrambi presenti

- Concatenare i due files

```
cat <file_1> <file_2> > newFile.txt
```
- Contare di quante righe è composto il nuovo file

```
wc -l newFile.txt
```

La freccia **REDIREZIONA** il risultato del comando "cat" (la concatenazione) in un nuovo file. E' il comando base che vi permetterà di stampare (e creare) su un file neo-creato molti dei risultati dei programmi che andremo ad utilizzare.

Come potete osservare, il nuovo elemento non pregiudica l'esistenza degli altri due che permangono nella cartella.

- rimuovere il nuovo file creato

La strategia che rende la shell una via estremamente rapida per gestire i files è la stessa che abbiamo accennato per quanto riguarda le directories.

Infatti, è possibile compiere “azioni a distanza” che rendono il tutto estremamente funzionale.

Esercizio: azioni sui files “a distanza”

Vedremo ora come eseguire le stesse azioni dell’esercizio precedente ma da una posizione diversa rispetto a quella dove avranno luogo.

- Entrare nella directory Desktop
- Dal desktop, creare il file di testo direttamente in /Home/NomeUtente/

```
nano /Home/NomeUtente/
```

spiegazione: nonostante l’editor di testo si sia aperto normalmente, in realtà non viene eseguito nel Desktop ma direttamente nella cartella /NomeUtente/

- Scrivere un’altra lista di 10-20 nomi e salvare il file.
- Dove è stato salvato il file?
- Dalla posizione in cui siete (Desktop), spostare il file nella Directory_2

```
mv /Home/NomeUtente/NewFile /Home/NomeUtente/Directory_1/Directory_2/
```

spiegazione: con questa operazione il file è stato spostato di directory. **TRA NewFile e /Home/... C’è UNO SPAZIO!**

- Dalla posizione in cui siete, concatenare i tre files in un unico file

```
cat /Home/NomeUtente/Directory_1/Directory_2/file_1 /Home/NomeUtente/  
Directory_1/Directory_2/file_2...E CONCLUDETE VOI
```

- Dalla posizione in cui siete, contare di quante righe è composto il file creato

Esercizio: azioni sui files “a distanza”: METTIAMO IN PRATICA IL TUTTO

- andare nella Directory_1 e creare un file di testo
- dalla Directory_1, creare un altro file di testo nella Directory_2
- muoversi nella Home
- dalla Home, muovere i due files nel Desktop e concatenarli
- Andare nella Directory_2 e da questa, contare quante righe ha il file creato nel Desktop