

Introduction à Linux RAQ - Cours 01

Eric Normandeau - 2015-02-02

(Version NON terminée)

Plan de cours

1. Introduction
2. Connection au serveur Linux
3. Terminal et commandes de base
4. Liste de commandes importantes

1 - Introduction

1.1 - À propos du cours

- Débute le 4 février
- Deux heures par semaine
- De 10h à 12h le mercredi matin
- Semi-magistral avec exercices (apportez votre laptop)
- N'hésitez pas à poser des questions durant le cours !
- Donné dans la salle Hydro-Québec à l'Université Laval (Québec)
- Disponible en direct avec Google Hangout
- Disponible en différé sur [YouTube](#)
- Notes de cours disponibles sur [GitHub](#)
- Utilisateurs Windows doivent télécharger [putty.exe](#)

1.2 - Qu'est-ce que Linux ?

Linux est un système d'exploitation, au même titre que Windows et MacOSX. Bien qu'il soit gratuit, Linux est beaucoup moins fréquemment retrouvé sur les ordinateurs standards (tours et ordinateurs portables). Cependant, dans le monde des ordinateurs très puissants (serveurs de calcul et super-calculateurs), près de 95% des systèmes installés sont dérivés de UNIX, dont la grande majorité sont des versions de Linux. Il existe plusieurs dizaines de versions différentes de Linux,

mais il y en a seulement 4-5 qui sont plus fréquentes et elles sont très similaires. Afin d'alléger la discussion, je vais parler de "systèmes UNIX" ou même de "Linux" pour désigner à la fois Linux et MacOSX. Il faut savoir cependant que bien que ces deux systèmes d'exploitation soient similaires, ils sont quand même légèrement différents et certains programmes ou commandes qui fonctionnent avec l'un ne fonctionnent pas avec l'autre.

Les systèmes UNIX descendent du système d'exploitation original UNIX à l'époque où les seuls ordinateurs se trouvaient dans les universités, les gouvernements et les très grosses compagnies. UNIX fait partie des tous premiers systèmes qui pouvaient être installés sur des ordinateurs fabriqués par différentes compagnies. Avant cette époque, chaque compagnie vendait son propre système d'exploitation et c'était le seul qui pouvait fonctionner sur leurs ordinateurs. UNIX a donc révolutionné la manière dont des chercheurs dans différentes universités pouvaient collaborer puisque le code qui fonctionnait sur une machine UNIX fonctionnait également sur une autre. Cette tradition s'est poursuivie au fil des années avec les descendants de UNIX, dont Linux est rapidement devenu le plus populaire après sa création au début des années 90. Linux est un logiciel libre et on peut obtenir la plupart des versions, appelées *distributions*, gratuitement.

1.3 - Pourquoi utiliser Linux en biologie ?

Les besoins de la recherche scientifique en général et de la biologie en particulier, s'allignent très bien avec les capacités offertes par les systèmes UNIX :

- Explorer rapidement des données et des hypothèses
- Reproduire des analyses de façon répétée sur différents jeux de données
- Analyser des données demandant une grande puissance de calcul ou beaucoup de mémoire

Les systèmes UNIX fournissent un environnement et des outils rendent ces tâches plus faciles. En utilisant Linux en combinaison avec un ou plusieurs langages de programmation, on peut plus rapidement explorer ses données, reproduire rapidement des analyses et lancer des programmes gourmands en ressources de calcul (nombre de cœurs et quantité de mémoire).

Dans certains domaines de la bioinformatique, on retrouve presque exclusivement des programmes pour Linux et MacOSX. Par exemple, la plupart des programmes pour faire des alignements de séquences et découvrir des SNPs sont conçus pour

être installés et utilisés sur Linux. Afin d'avoir accès à ces programmes, il faut donc savoir utiliser des ordinateurs avec des systèmes UNIX.

1.4 - Exemples d'analyses

Analyse de séquences

- Assemblage et annotation de génomes
- Génotypage d'individus par Séquençage
- Trouver des séquences similaires
- Alignement de séquences

Expression de gènes

- Biopuces (*tellement* 2008)
- RNA-Seq (séquençage d'ADNc)
- Analyse de la régulation d'expression

Structure de protéines

Durant la formation, nous allons tenter de faire de courtes analyses avec des séquences d'ADN :

- Compter des séquences
- Nettoyer des fichiers de séquences
- Rechercher des séquences similaires avec **blast**
- Assembler des génomes bactériens

2 - Connection au serveur Linux

2.1 - Pré-requis

- Sur Linux et MacOSX: ouvrir une fenêtre de terminal.
- Sur Windows, télécharger [putty.exe](#), sauvegarder sur votre bureau.

2.2 - Connexion sur Linux ou MacOSX

Afin de se connecter au serveur, nous devons premièrement ouvrir un terminal. Pour les utilisateurs de MacOSX et Linux, lancez la commande suivante dans le terminal que vous avez ouvert:

```
ssh <username>@raq.ibis.ulaval.ca
```

Vous devez remplacer **<username>** par le nom d'utilisateur que vous avez reçu. Lorsque demandé, entrez votre mot de passe. Il est normal que vous ne voyez pas d'étoiles ou de cercles apparaître lorsque vous tapez votre mot de passe.

2.3 - Connexion sur Windows

Double-cliquer sur l'exécutable **putty.exe** que vous avez sauvegardé sur votre bureau.

Dans la case **Host Name (or IP address)**, tapez le nom du serveur :

```
raq.ibis.ulaval.ca
```

Cliquez sur le bouton **Open** en bas. Un terminal s'ouvrira où vous devrez entrer votre nom d'utilisateur (par exemple, **user_99**) et appuyer sur la touche **Enter**. Lors de la première connection, **putty.exe** vous avertira que vous ne connaissez pas le serveur et vous demandera de confirmer que vous souhaitez poursuivre. Cliquez sur **Yes** pour continuer. Le serveur vous demandera alors de taper votre mot de passe et d'appuyer sur la touche **Enter**. C'est normal si aucun caractère n'apparaît pendant que vous tapez votre mot de passe. Après avoir appuyé sur la touche **Enter**, vous serez connecté au serveur du cours.

Nous allons maintenant configurer **putty.exe** pour que l'affichage soit plus plaisant. Si vous êtes connecté, tapez **exit** dans le terminal puis appuyez sur **Enter**.

Relancez **putty.exe** au besoin. Cliquez sur l'onglet **Appearance** à gauche. Cliquez sur l'option **Antialiased** à peu près au centre dans la section **Font Settings**. Toujours dans la même section, cliquez sur le bouton **Change...** pour modifier la police de caractères. Choisissez une police qui contient le mot **Mono** dedans, ajustez la taille à 11 ou 12, et appuyez sur **OK**. Retournez dans l'onglet **Session** en haut à gauche et entrez le nom du serveur (**raq.ibis.ulaval.ca**) et nommez la session **RAQ** (au centre). Cliquez maintenant sur **Save** pour sauvegarder les options que nous venons de choisir. Vous êtes maintenant prêts à utiliser **putty.exe**.

3 - Terminal et commandes de base

Si vous avez suivi les étapes décrites à la section 2, vous êtes maintenant connecté à un serveur Linux et vous regardez un terminal vide avec un curseur qui attend vos commandes.

Le terminal est une fenêtre interactive où on tape des commandes et où les résultats peuvent être affichés. Il existe plusieurs types de terminaux, mais le terminal **bash** est de loin le plus fréquent. Nous allons utiliser le terminal **bash** tout au long du cours. Il faut bien comprendre qu'**on ne peut pas utiliser Linux pour des analyses si on ne comprend pas bien comment le terminal bash et le système fonctionnent**. Notre but ultime reste de faire de la biologie, mais afin de pouvoir faire certaines de nos analyses sous Linux ou MacOSX, nous devons bien comprendre les bases du terminal.

Dans cette section, nous allons commencer à apprendre les commandes de base du terminal. Le terminal **bash** est très complet. Il permet de manipuler fichiers et dossier, installer des programmes, éditer des fichiers de texte, manipuler du texte, lancer des programmes. Il possède même son propre langage de programmation, dont nous allons seulement survoler les bases les plus utiles.

3.1 - Comment lancer des commandes

Pour utiliser une commande, on la tape dans le terminal et on appui sur **Enter**. Par exemple, pour afficher la date, on peut utiliser la commande **date** :

```
date
```

Pour afficher du texte à l'écran, on utilise la commande **echo** :

```
echo "Bienvenu à la formation Linux"
```

3.2 - Copier-coller dans le terminal

Si vous souhaitez copier-coller des commandes dans le terminal sous MacOSX et Linux, il se peut que vous ayez à utiliser des touches différentes de **Ctrl-C** et **Ctrl-V**. Par exemple, sur MacOSX, vous aurez peut-être à utiliser **Cmd-C** et **Cmd-V**, alors que sur Linux, vous utiliserez **Ctrl-Shift-C** et **Ctrl-Shift-V**.

3.3 - Commandes de base

Il existe des commandes de base qui permettent d'explorer et de manipuler des fichiers et des dossiers. Nous allons commencer à apprendre les plus communes aujourd'hui.

Une fois connecté, nous voulons savoir où nous sommes. La commande **pwd** (pour *present working directory*) nous indique le chemin (path) où nous nous trouvons :

```
pwd
```

Nous sommes dans le dossier **/home/username** où la partie *username* est votre nom d'utilisateur.

Pour voir ce qu'il y a dans le dossier, nous utilisons la commande **ls** :

```
ls
```

La commande nous indique la présence d'un fichier nommé **mot_de_bienvenue.txt**. Nous allons afficher le message contenu dans le fichier avec la commande **cat** :

```
cat mot_de_bienvenue.txt
```

On peut utiliser la touche **Tab** pour compléter automatiquement les noms de commandes, fichiers et dossiers lorsqu'il n'y a pas d'ambiguïté possible. Testez-le en tapant **cat mot<tab>**. Le nom du fichier devrait se compléter automatiquement. Cette façon d'écrire les noms de fichiers est préférée car elle sauve du temps et évite des erreurs de frappe.

Afin de pouvoir tester les autres commandes, nous allons avoir besoin de plus de dossiers et de fichiers. Nous allons donc copier un dossier déjà préparé pour le cours 01 avec la commande **cp**, que nous allons revoir plus tard :

```
cd # Pour retourner dans notre dossier d'utilisateur  
cp /cours_intro_linux/cours_01 .
```

Nous pouvons vérifier que le dossier a été copié avec **ls** :

```
ls
```

Avec les bonnes options, on pourrait afficher les fichiers et dossiers comme une liste et voir leur taille. L'option **-l** affiche les fichiers en liste et l'option **-h** donne leur taille de façon plus lisible (en Ko, Mo, Go...) :

```
ls -lh
```

On pourrait même voir les fichiers cachés (ils commencent par un point **.**) en ajoutant l'option **-a** (pour *all*), qui affiche tout :

```
ls -lha
```

Pour se déplacer dans les dossiers, on utilise la commande **cd**. Par exemple, pour aller dans le nouveau dossier **cours_01** et lister le contenu du dossier:

```
cd cours_01
ls -lh
```

On voit qu'il y a un dossier **README.txt**. Pour le lire, on utilise encore la commande **cat** :

```
cat README.txt  # N'oubliez pas d'utiliser <tab>
```

Nous allons nous déplacer dans le dossier **00_alice** et regarder le contenu :

```
cd 00_alice
ls
```

On découvre un fichier nommé **alice.txt**. Nous allons tenter de le visualiser en utilisant les commandes **cat**, **head**, **tail** et **less** : “

```
wc -l alice.txt  # Nombre de lignes dans le fichier

cat alice.txt    # Pas très intéressant
head alice.txt   # Pour voir le début du fichier
tail alice.txt   # Pour voir la fin du fichier

less alice.txt   # Pour lire tout le fichier
```


Nous allons maintenant créer, déplacer, renommer et effacer des fichiers et des dossiers. Pour cela, nous allons nous déplacer dans le dossier `~/cours_01/02_fichiers_et_dossiers`. Chaque fois que vous voyez le symbole `~` dans un chemin de dossier, il fait référence à votre dossier d'utilisateur dans `/home/username`. Par exemple, pour l'utilisateur **eric**, la commande suivante retourne dans `/home/eric`, se déplace dans le dossier `cours_01/02_fichiers_et_dossiers` qui s'y trouve et en liste le contenu:

```
cd ~  
cd cours_01/02_fichiers_et_dossiers  
ls -lh
```

Pour copier des fichiers, on utilise la commande **cp** suivie du nom du fichier à copier et puis du nom du nouveau fichier où le copier. Afin de copier un dossier, il faut utiliser l'option **-r** :

```
cp fichier_01.txt copie_fichier_01.txt  
cp -r dossier_01 copie_dossier_01  
  
ls -lh
```

Pour déplacer ou renommer des fichiers ou dossiers, on utilise la commande **mv** :

```
mv copie_fichier_01.txt fichier_02.txt  
mv copie_dossier_01 dossier_02  
  
ls -lh
```

Finalement, pour effacer un fichier ou un dossier, on utilise la commande **rm**. Pour les dossiers, il faut ajouter l'option **-r**, comme pour la commande **cp**.

```
rm fichier_02.txt  
rm -r dossier_02  
  
ls -lh
```

Pour créer des fichiers et des dossiers vides, on utilise les commandes **touch** et **mkdir** :

```
touch nouveau_fichier.txt  # Extension .txt pas nécessaire  
mkdir nouveau_dossier  
  
ls -lh
```

4 - Liste de commandes importantes

Voici une courte liste des commandes que nous avons utilisé aujourd'hui :

Explorer des dossiers

- **ls** : Lister les fichiers et les dossiers (list) [-l -h]
- **pwd** : Afficher le dossier courant (present working directory)
- **cd** : Changer de dossier (change directory) [. .. ~ -]

Explorer des fichiers

- **wc** : Afficher le nombre de lignes d'un fichier [-l]
- **cat** : Afficher le contenu d'un fichier
- **head** : Afficher le début d'un fichier [-n]
- **tail** : Afficher la fin d'un fichier [-n]
- **less** : Lire un fichier

Créer, copier et déplacer des fichiers et dossiers

- **cp** : Copier des fichiers et dossiers (copy) [-r]
- **mv** : Déplacer ou renommer fichiers et dossiers (move)
- **rm** : Effacer des fichiers ou des dossiers (remove) [-r -f]
- **touch** : Créer (ou mettre à jour) un fichier
- **mkdir** : Créer un dossier

3.4 Autres commandes utiles

- **history** : Lister les commandes utilisées
- **man** : Lire le manuel d'une commande
- **echo** : Afficher du texte à l'écran
- **clear** : Vider le terminal (aussi **Ctrl-L**)
- **sleep** : Attendre pour un laps de temps (secondes, minutes...)
- **Ctrl-C** : Canceller une commande
- **Ctrl-D** : Sortir d'un programme ou arrêter le terminal

Pour prochain cours

`cd . .. ~ - /` alias (eg: `ll`) text file formats (Windows `\n`, MacOS `^M`, Linux `\n`) Sous Linux, la racine absolue du système est `/`, appelé *slash*. `grep -c ">" fichier.fasta` # Attention aux guillemets!!!

Explorer les séquences

Nous allons tester ces mêmes commandes avec le dossier de séquences. Premièrement, il faut remonter au dossier supérieur `..`, trouver le dossier de séquences et s'y déplacer :

```
cd ..  
ls  
  
cd 01_sequences  
ls
```

On voit qu'il y a deux fichiers des séquences **sequences_01.fasta** et **sequences_02.fasta**. On va utiliser les commandes **wc -l**, **head**, **tail** et **less** pour les explorer.

```
wc -l sequences_01.fasta  
wc -l sequences_02.fasta  
  
# Ou, encore mieux  
wc -l *.fasta # * veut dire n'importe quelle chaîne de texte  
  
# Voir le début des fichiers  
head sequences_01.fasta  
head sequences_02.fasta  
  
# Ou, encore mieux  
head -n 6 *.fasta
```