

Session de formation 2018

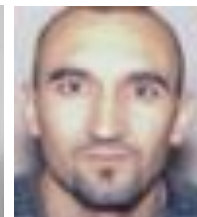
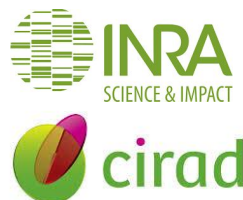


- 12 Mars** **Guide de survie à Linux : les commandes de base pour débiter sur un serveur linux**
- 13 Mars** Linux avancé : manipuler et filtrer des fichiers sans connaissance de programmation
- 15 Mars** Initiation à l'utilisation du cluster bioinformatique itrop
- 22 Mars** Initiation à git
- 23 Mars** Initiation aux gestionnaires de workflow South Green: Galaxy ou TOGGLE
- 26 Mars** Initiation aux analyses de données transcriptomiques



South Green

bioinformatics platform





Institut de Recherche
pour le Développement

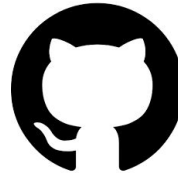
South Green
bioinformatics platform



plateau i-trop



www.southgreen.fr



<https://github.com/SouthGreenPlatform>



The South Green portal: a comprehensive resource for tropical and Mediterranean crop genomics, Current Plant Biology, 2016

Session de formation 2018



- Toutes nos formations :
<https://southgreenplatform.github.io/trainings/>
- Topo & TP : [Linux For Dummies](#)
- Environnement de travail : [Logiciels à installer](#)

Guide de survie à Linux



www.southgreen.fr

<https://southgreenplatform.github.io/trainings>



The objectif!

Lancez vos analyses bioinformatiques sous Linux !



Applications

Connaître les principales commandes sous Linux

- Se déplacer dans l'arborescence de fichier : *pwd*, *ls*, *cd*, *mkdir* etc.
- Se connecter à un serveur, transférer les données : *ssh*, *scp*, *wget*
- Manipuler des fichiers : *head*, *tail*, *sort*, *cut*, *wc*
- Lancer des logiciels bioinformatiques en ligne de commande



Introduction

- **Systeme d'exploitation** réputé pour :
 - sa sécurité
 - ses mises à jour fréquentes
 - son prix et ses programmes gratuits

- Créé en 1991 par **Linus Torvalds**

- Basé sur l'OS propriétaire Unix (1969)

- Linux **gratuit** et **libre**

on peut avoir le code source, la “recette de fabrication”
on peut copier, modifier, redistribuer



- **Système robuste et multi-plateforme**

- **Système multi-utilisateurs**

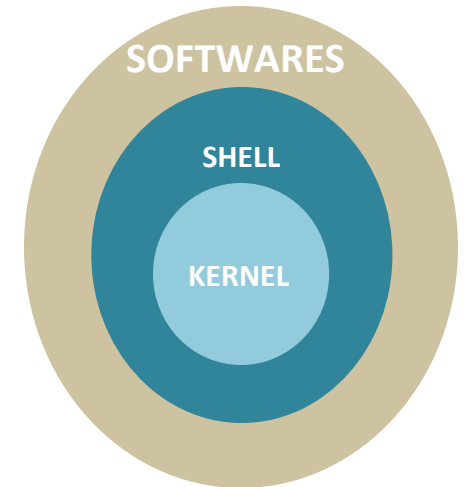
Plusieurs utilisateurs peuvent travailler en même temps

- **Système multi-tâches (processus/programmes)**

Chaque utilisateur peut lancer plusieurs programmes en même temps



Distribution : Noyau/Coeur + logiciels/programmes



- 2 façons d'utiliser linux :

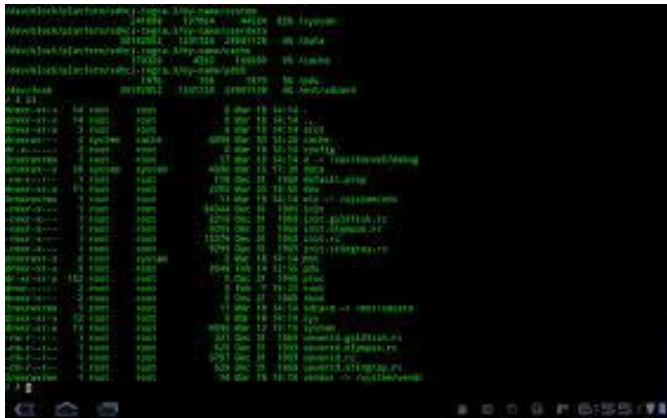
en *mode graphique*



- 2 façons d'utiliser linux :

en *mode graphique*

en *mode console*



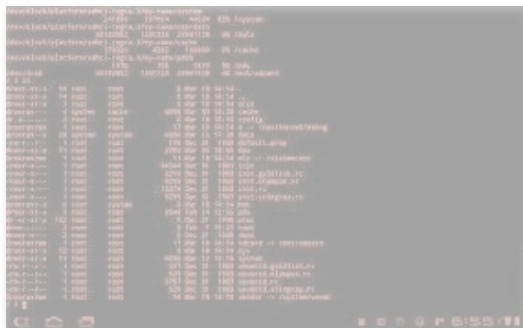


- Nombreux programmes rapides & puissants
- Facile de lier des commandes/programmes entre eux (workflow)
- Nombreux outils bioinformatique disponibles
- Pas besoin de ressources matérielles importantes
- 90% des serveurs fonctionnent sous Linux

Pourquoi utiliser Linux ?



- Nombreux programmes rapides & puissants
- Facile de lier des commandes/programmes entre eux (workflow)
- Nombreux outils bioinformatique disponibles
- Pas besoin de ressources matérielles importantes
- 90% des serveurs fonctionnent sous Linux



**Pas d'interfaces
graphiques**

**Convivialité de la ligne
de commande ?**





Nécessité de la pratique et de l'expérience

⇔ Investissement non négligeable pour de bons résultats rapidement

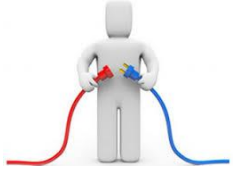


Environnement de travail

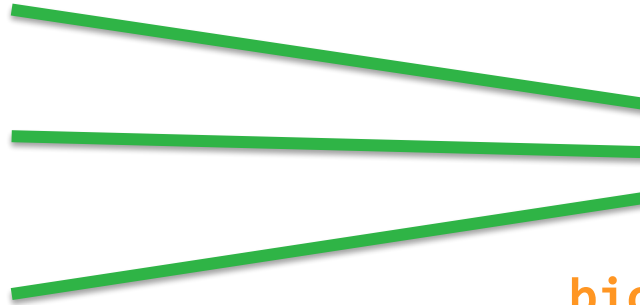
*Comment copier un fichier de son PC sur
le serveur ?*



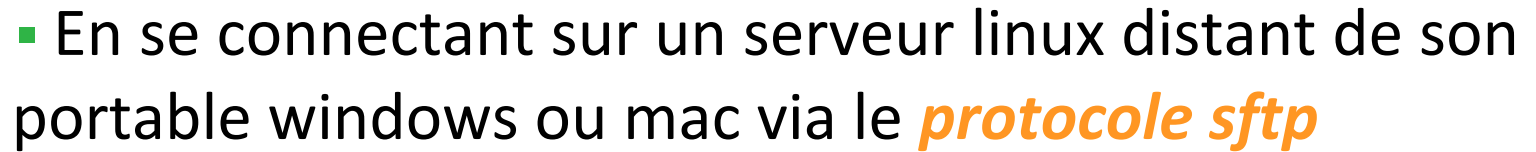
Copier un fichier de son PC sur le serveur ?



- En se connectant sur un serveur linux distant de son portable windows ou mac via le *protocole sftp*



bioinfo-nas.ird.fr





Practice

filezilla, sftp

1

Go to [Practice 1](#) on our github

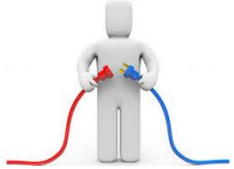


Environnement de travail

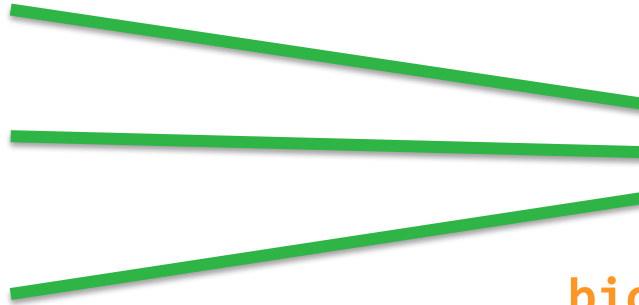
Comment travailler sur le serveur ?



Comment travailler sur le serveur ?



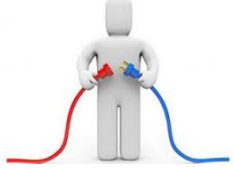
- En se connectant sur un serveur linux distant de son portable windows ou mac via le **protocole ssh**



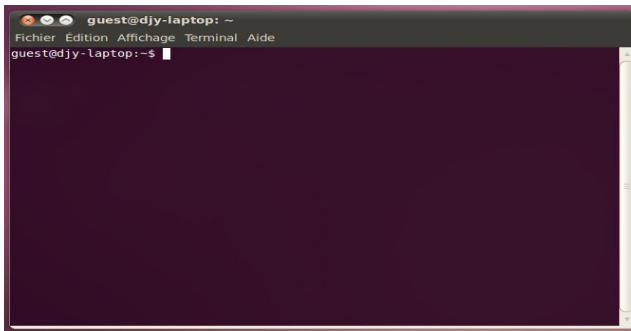
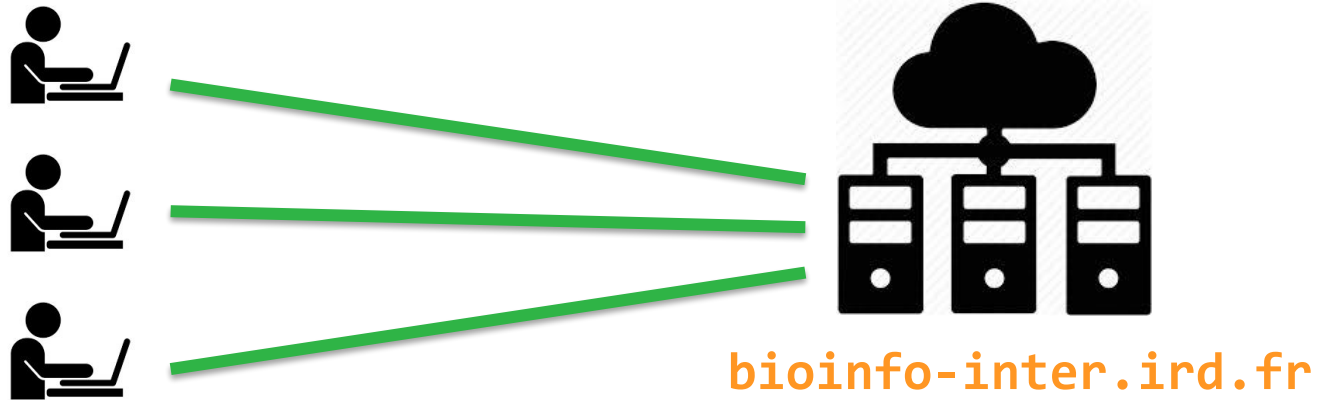
bioinfo-inter.ird.fr



Comment travailler sur le serveur ?



- En se connectant sur un serveur linux distant de son portable windows ou mac via le *protocole ssh*



PuTTY

ssh





Practice

putty,
terminal, ssh

2

Go to [Practice 2](#) on our github



Premiers Pas sur Linux

Commandes relatives à l'arborescence de
fichiers

Toujours présent sur le terminal, juste avant de taper la commande

Prompt `[tranchant@node6 data]$`

Toujours présent sur le terminal, juste avant de taper la commande

Prompt

```
[tranchant@node6 data]$
```

Nom
utilisateur

Nom
serveur

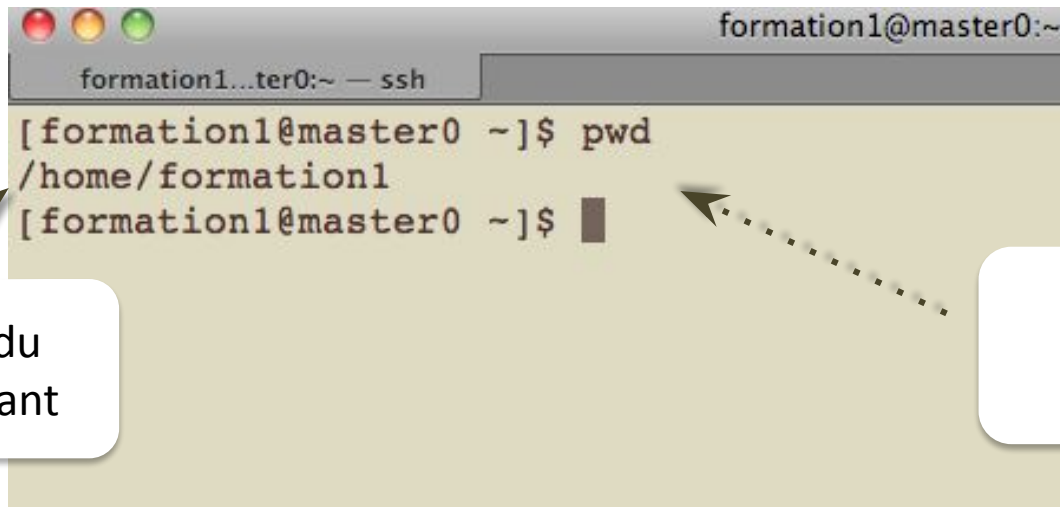
Répertoire
courant

commande [-options] [arguments]

pwd

Present Work Directory

affiche nom du répertoire courant (et son chemin complet)



```
formation1@master0:~  
[formation1@master0 ~]$ pwd  
/home/formation1  
[formation1@master0 ~]$
```

Affiche le nom du
répertoire courant

Commande
sans options et
sans argument

ls
list

liste fichiers & répertoires présents dans le répertoire courant



```
formation1@master0:~ — ssh — 97x37
formation1...ter0:~ — ssh
[formation1@master0 ~]$ ls
data  scripts
```

The image shows a terminal window with a title bar indicating an SSH session to 'master0'. The prompt is '[formation1@master0 ~]\$'. The user has entered the command 'ls', and the output shows two items: 'data' (in blue) and 'scripts' (in blue). Dotted arrows point from the text boxes below to the output of the command.

Liste des fichiers du répertoire courant (par défaut)

Commande sans options et sans argument

ls -l
list long

liste les fichiers avec des informations pour chaque fichier/répertoire

Commande avec l'option l et le nom d'un répertoire en argument

```
[formation1@master0 ~]$ ls -l /home/
total 312
drwx-----  6 abate      sat      4096 12 mars   2012 abate
drwx-----  5 adam       ggr      4096 23 mars   2012 adam
drwx----- 31 admin     admin    4096  3 août   11:35 admin
drwx-----  9 alizon    ete      4096 21 août   14:23 alizon
drwx----- 12 alvaro-wis effecteurs 4096 17 juin   16:19 alvaro-wis
drwx-----  4 auguy     rhizogenesis 4096  2 mars   2012 auguy
drwx-----  5 ayoubas   team1    4096 13 avril  2012 ayoubas
drwx-----  5 beule     bdp      4096  8 oct.   17:49 beule
drwx-----  9 bouniol   ggr      4096  2 oct.   15:00 bouniol
drwx----- 10 castillo  bdp      4096 10 oct.   15:55 castillo
```

liste détaillée des fichiers

Comment obtenir de l'aide sur une commande?

- avec l'option *--help* ou *-h* *ls --help* *ls -h* *blastn -h*
- avec la commande *man* *man ls*

Arborescence linux

pwd

Affiche le chemin absolu

ls

Liste tous les fichiers/répertoires

ls -l

Affiche toutes les informations sur les fichiers



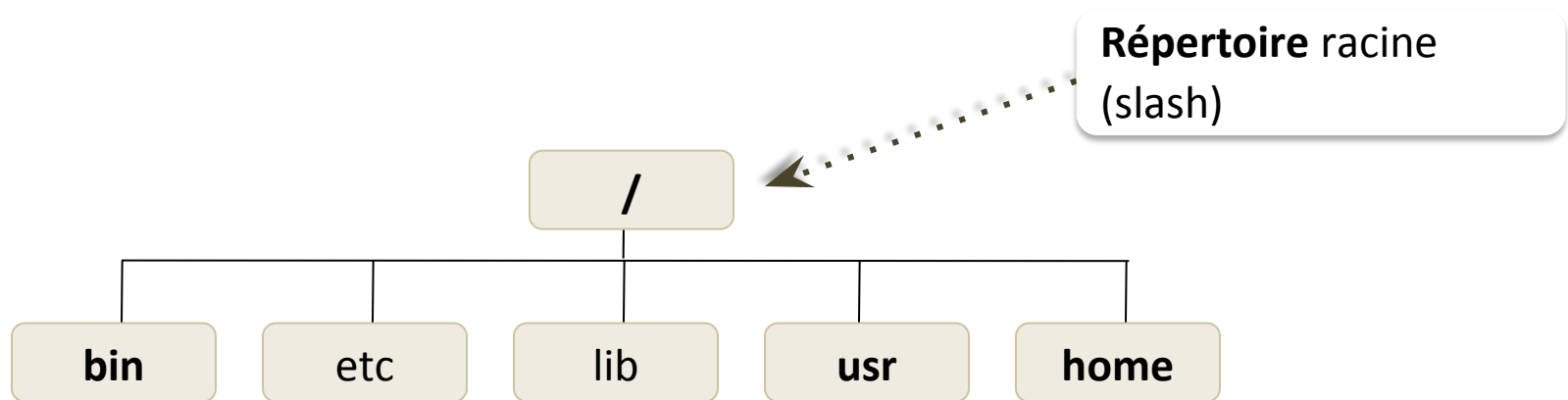
Practice

prompt, pwd

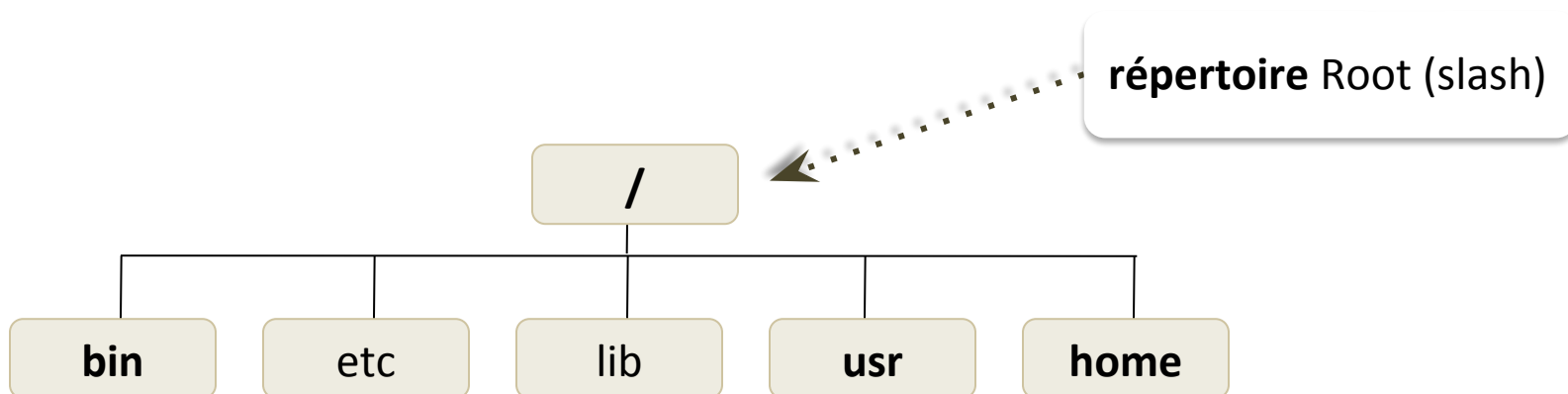
3

Go to [Practice 3](#) on our github

- Arborescence = système de fichiers
- Répertoire racine appelé “/”



Principaux répertoires



/bin commandes principales

/usr, /opt Applications et librairies

/usr/bin Autres commandes

/home répertoire des utilisateurs (1 par utilisateur/login)

Chemin (path) : chemin d'accès à un fichier/répertoire

Chemin (path) : chemin d'accès à un fichier/répertoire

absolu

- chemin complet du fichier en partant du répertoire racine /

Chemin (path) : chemin d'accès à un fichier/répertoire

absolu

- chemin complet du fichier en partant du répertoire racine /
- *commence toujours par /*
- Toujours correct peut importe où on l'on travaille

Chemin (path) : chemin d'accès à un fichier/répertoire

absolu

- chemin complet du fichier en partant du répertoire racine /
- commence toujours par /
- Toujours correct peut importe où on l'on travaille

relatif

- chemin défini par rapport où on est dans l'arborescence

Chemin (path) : chemin d'accès à un fichier/répertoire

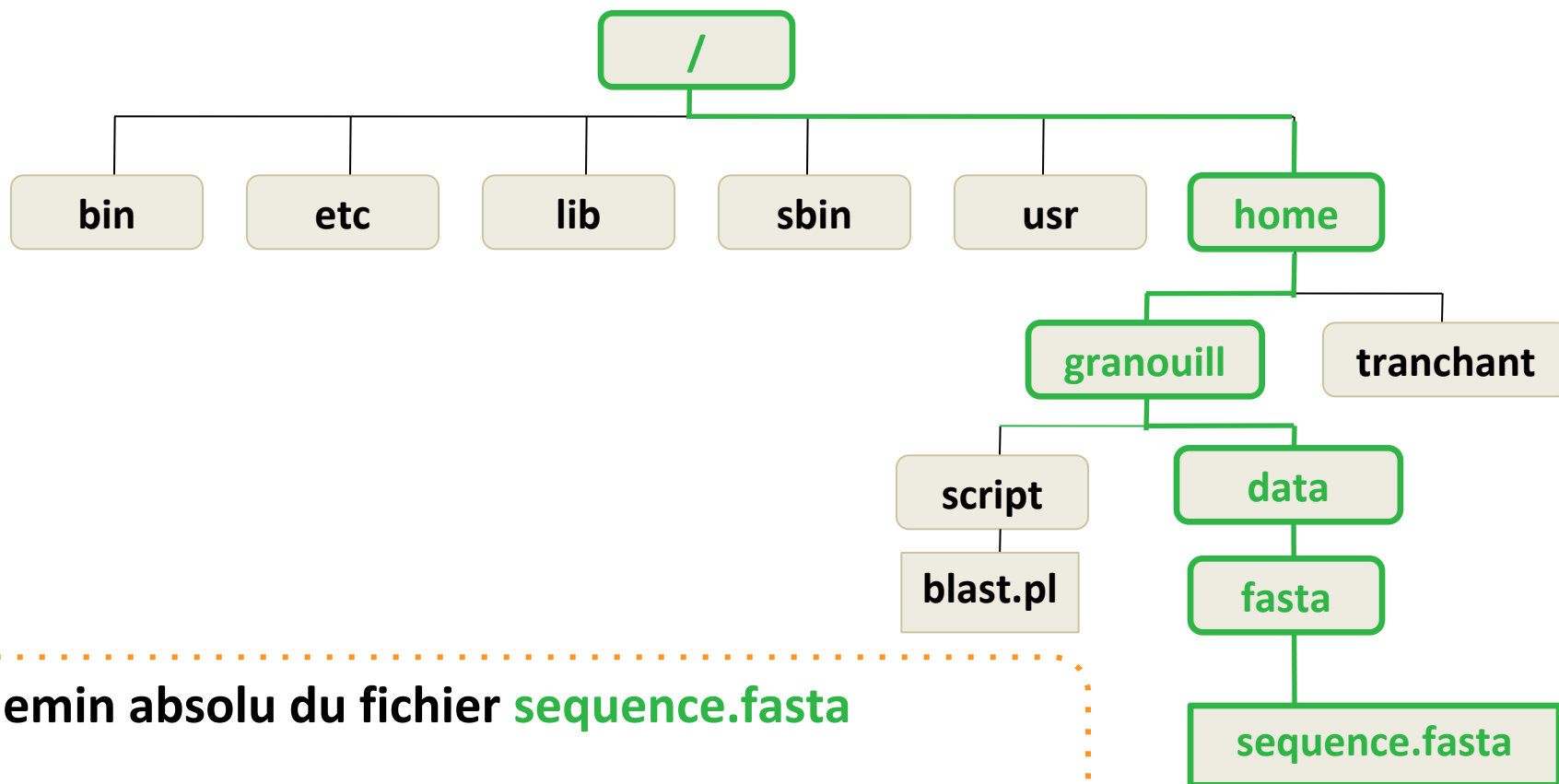
absolu

- chemin complet du fichier en partant du répertoire racine /
- commence toujours par /
- Toujours correct peut importe où on l'on travaille

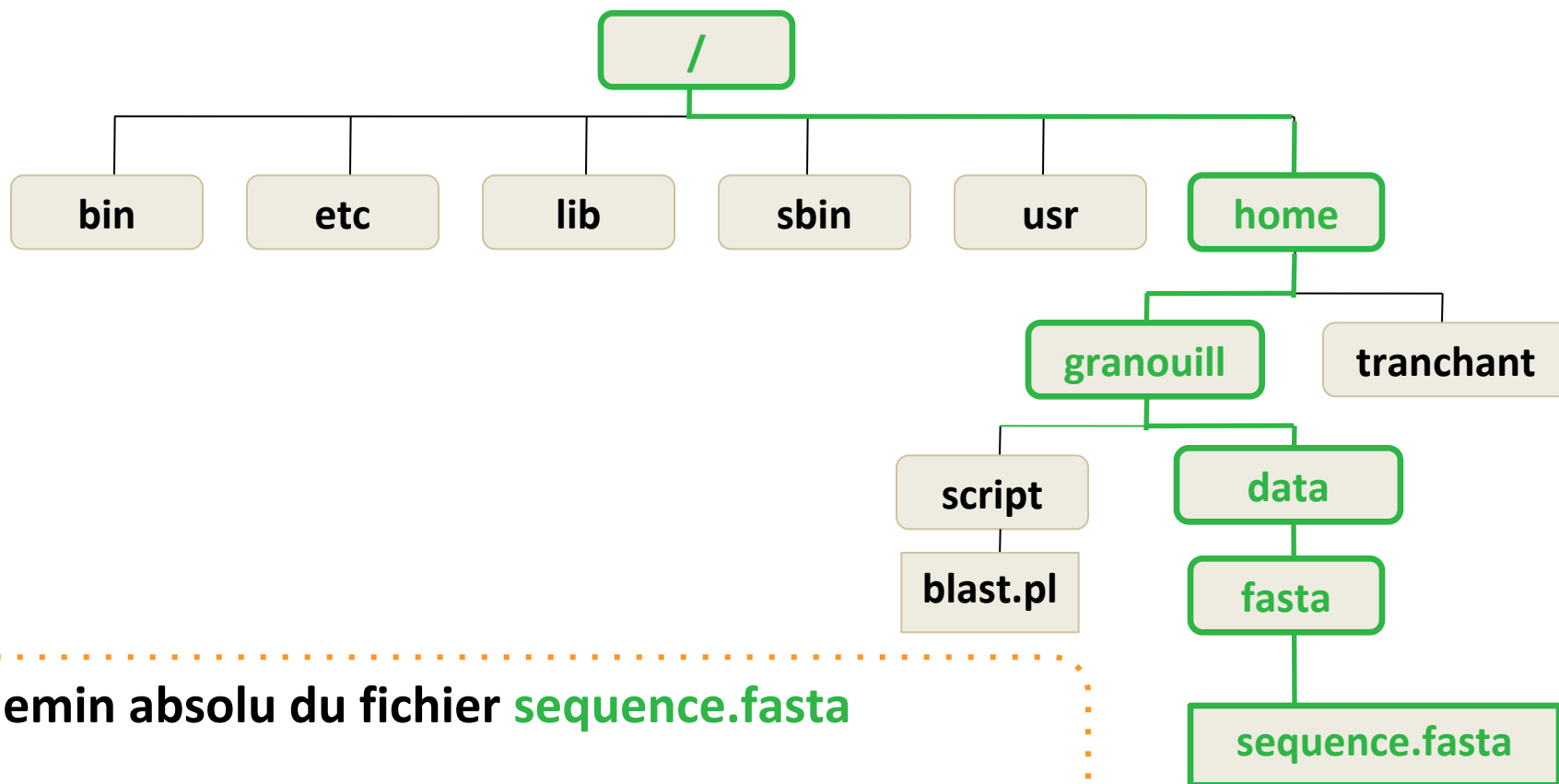
relatif

- chemin défini par rapport où on est dans l'arborescence
- *Ne commence jamais par /*
- **Change selon on l'on travaille**

- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où on l'on travaille



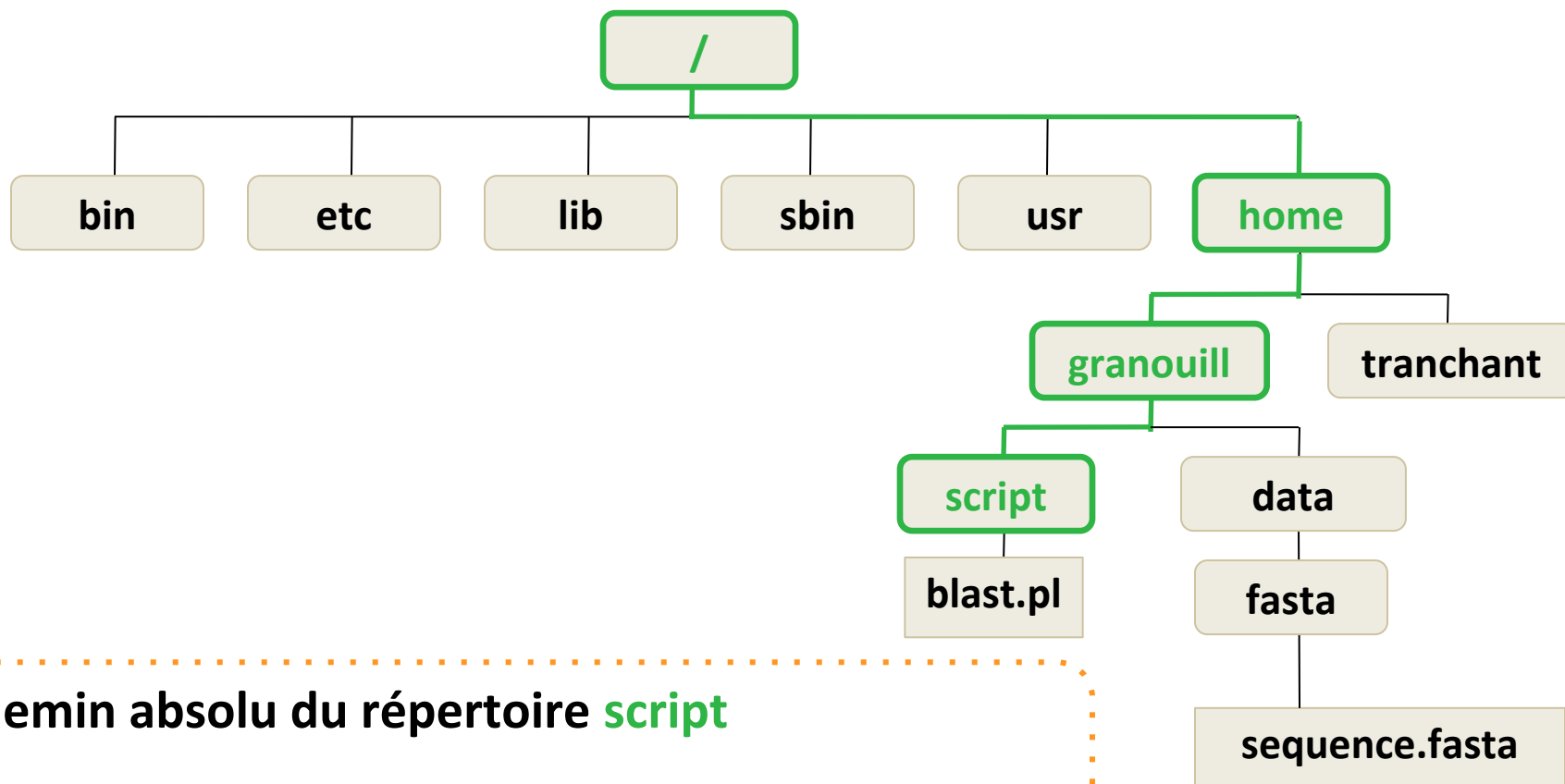
- Commence toujours par `/` (le répertoire racine)
- Toujours correct peu importe où on l'on travaille



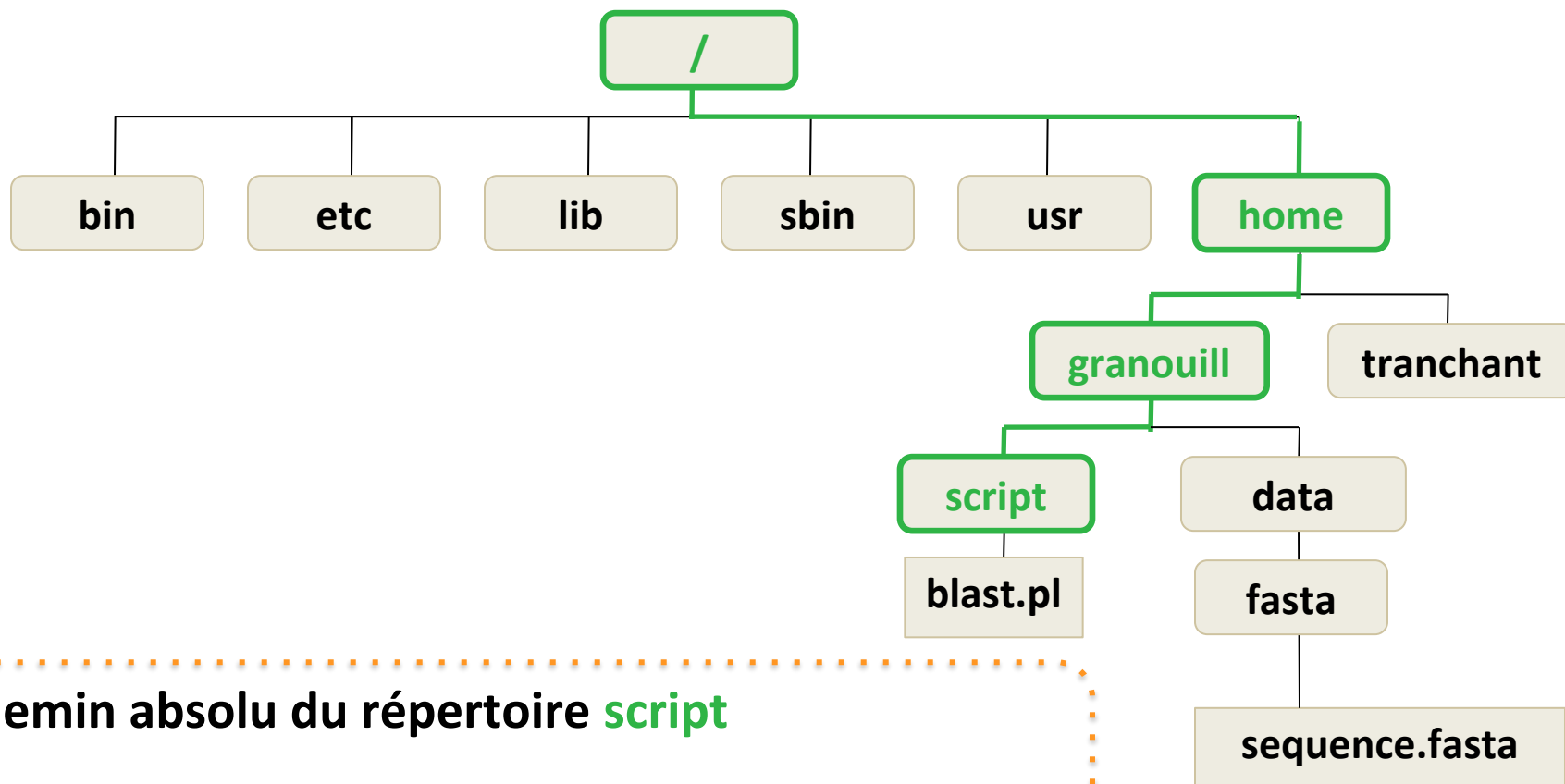
Chemin absolu du fichier **sequence.fasta**

`/home/granouill/data fasta/sequence.fasta`

- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où on l'on travaille



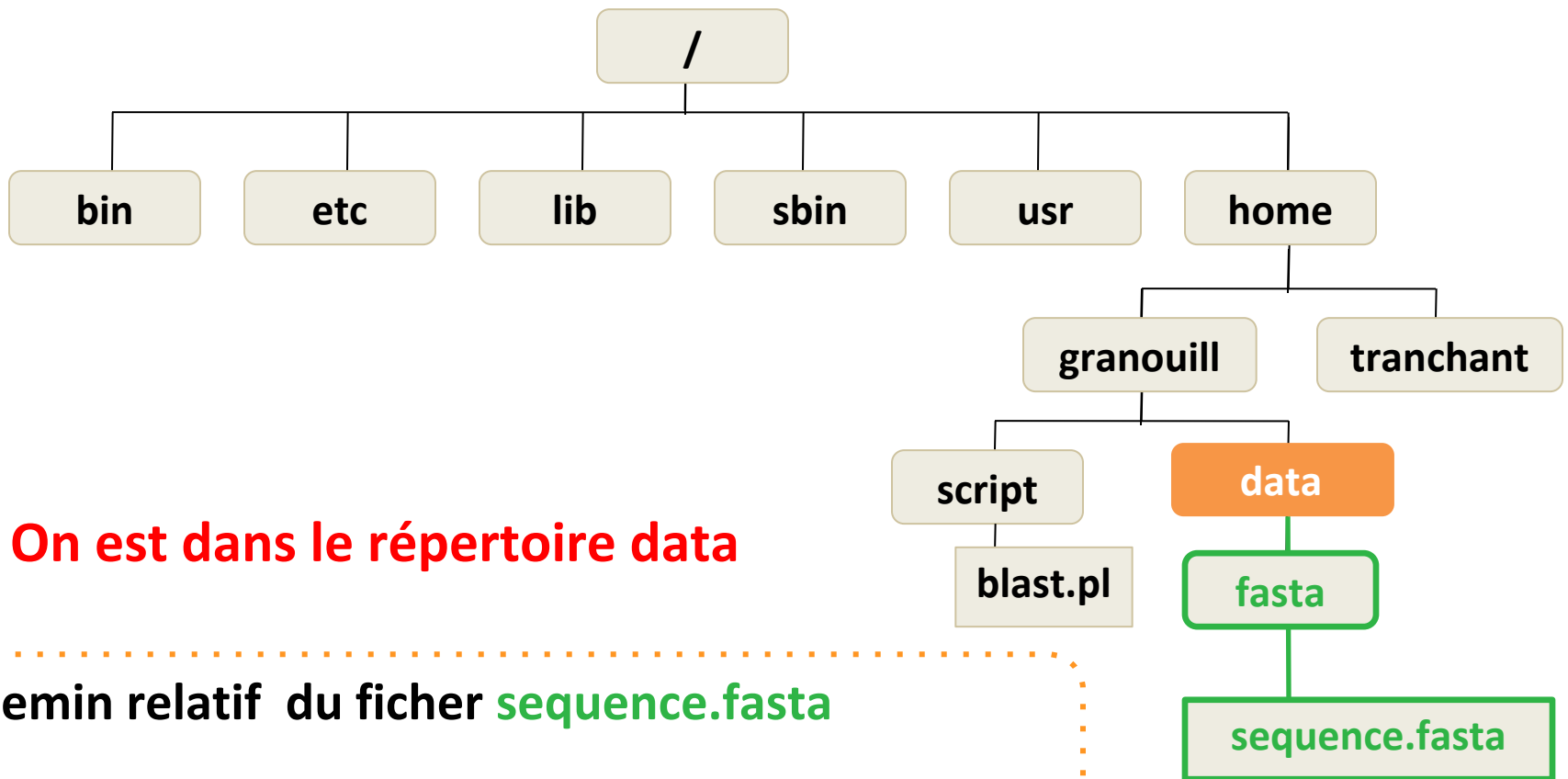
- Commence toujours par / (le répertoire racine)
- Toujours correct peu importe où on l'on travaille



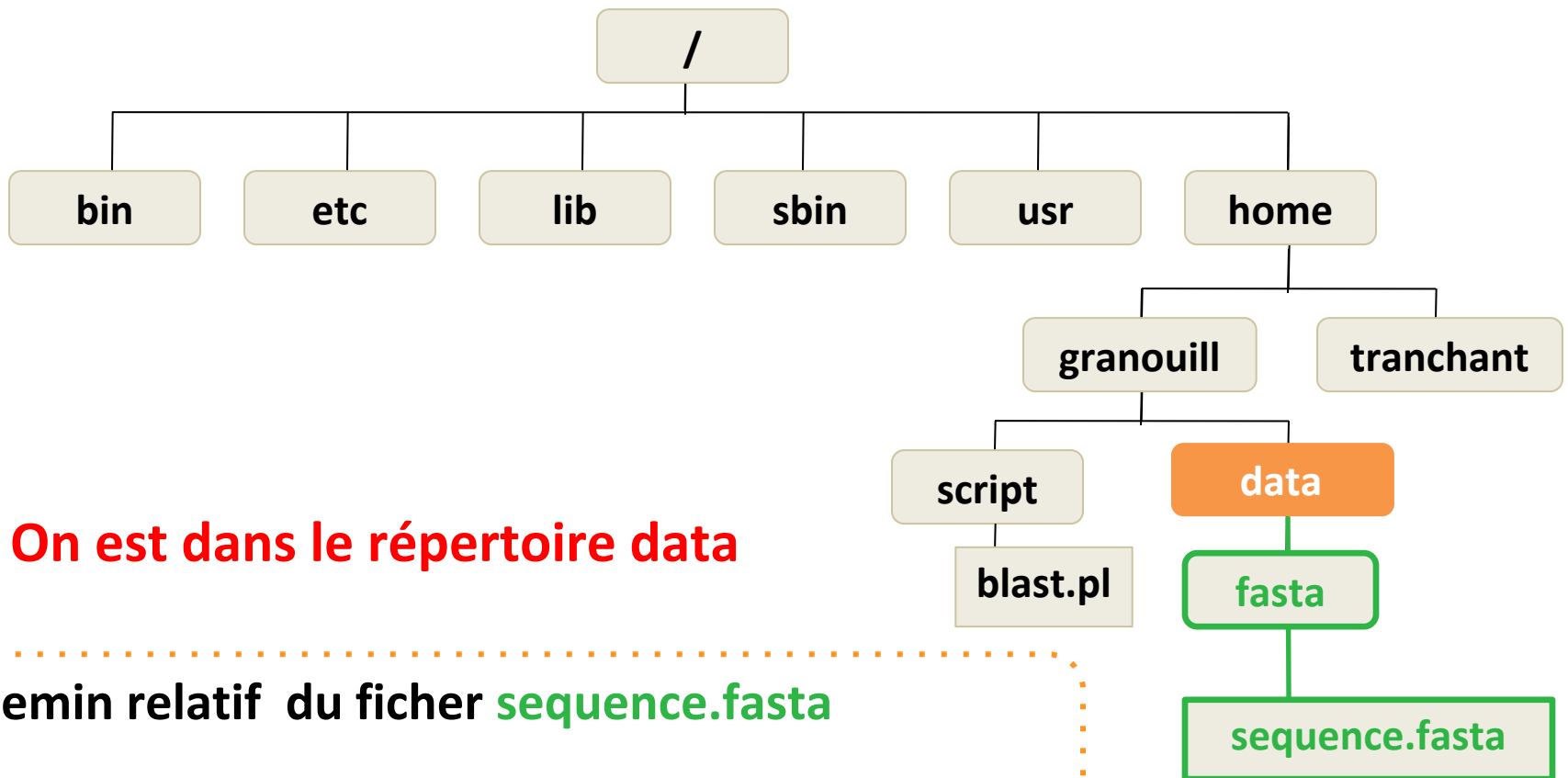
Chemin absolu du répertoire **script**

`/home/granouill/data/script`

- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**

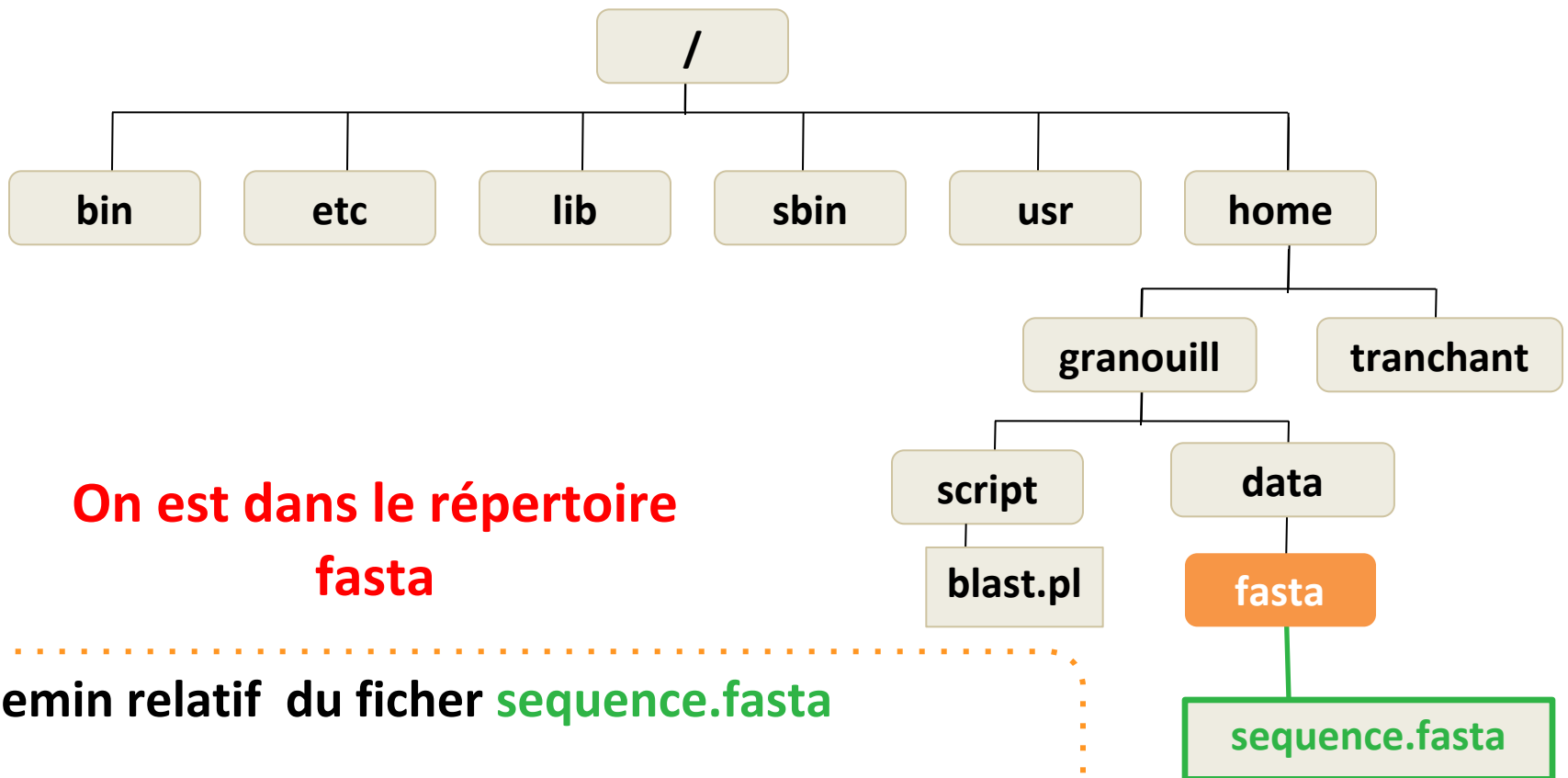


On est dans le répertoire data

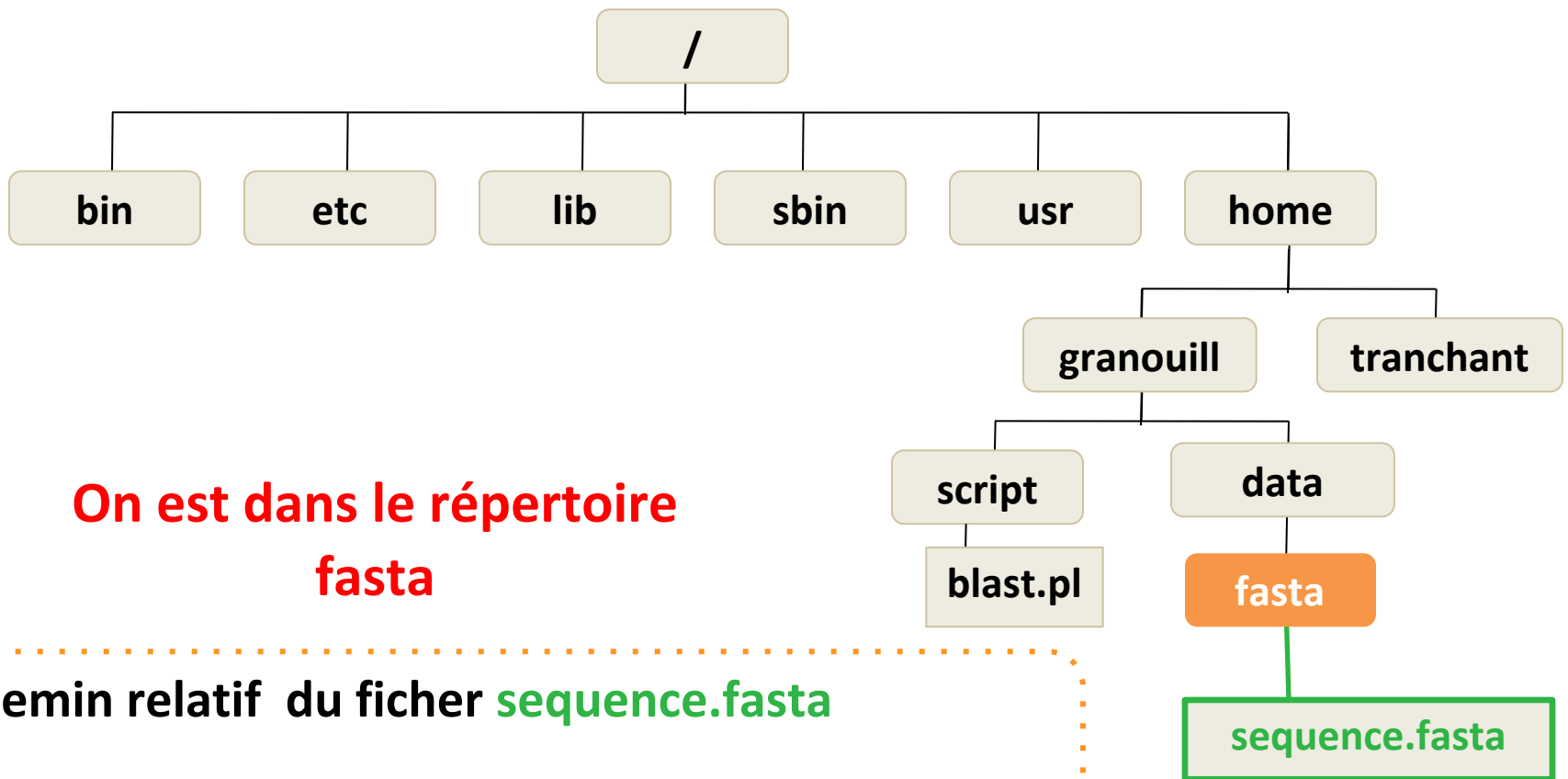
Chemin relatif du fichier **sequence.fasta**

fasta/sequence.fasta

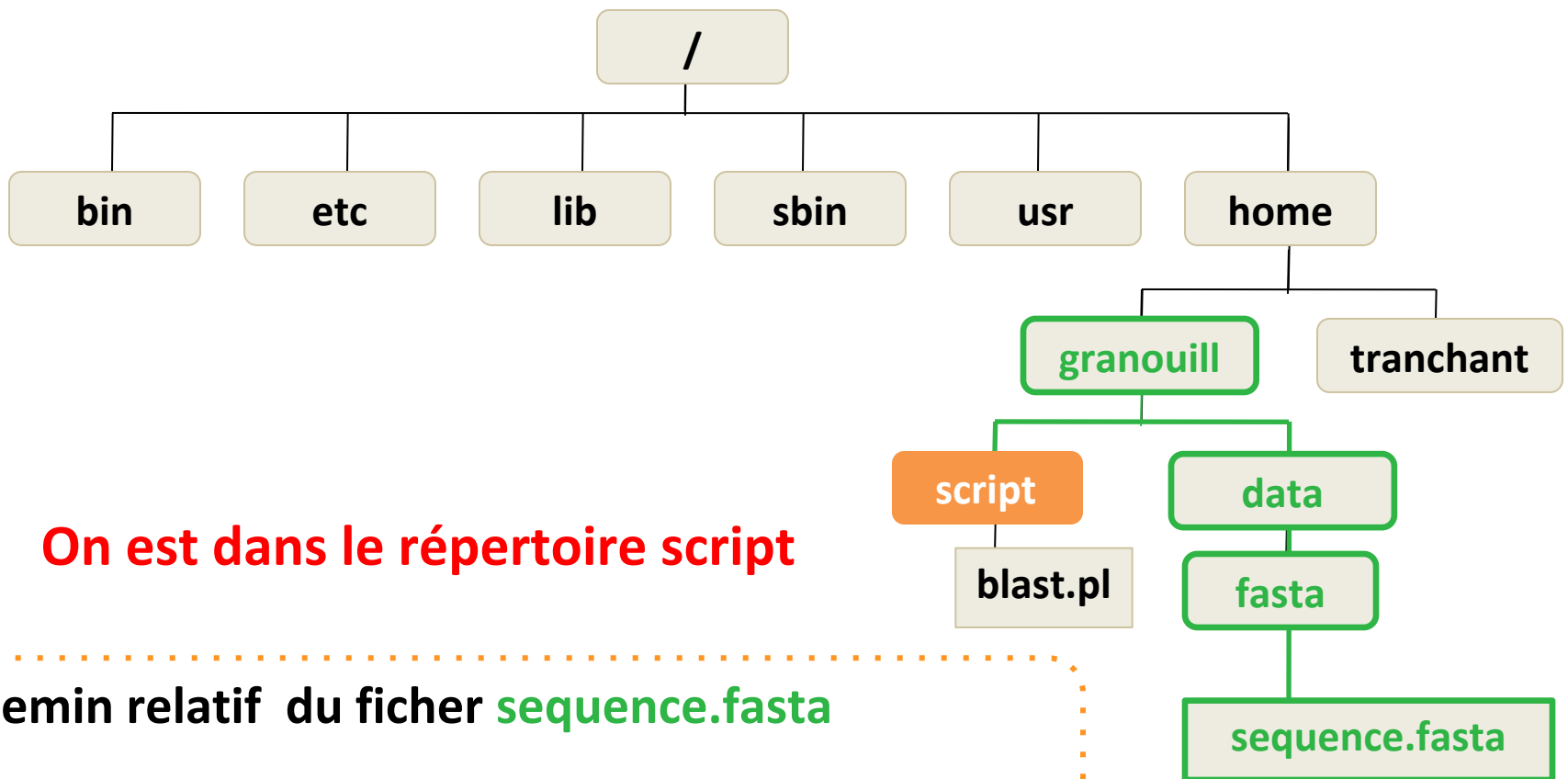
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



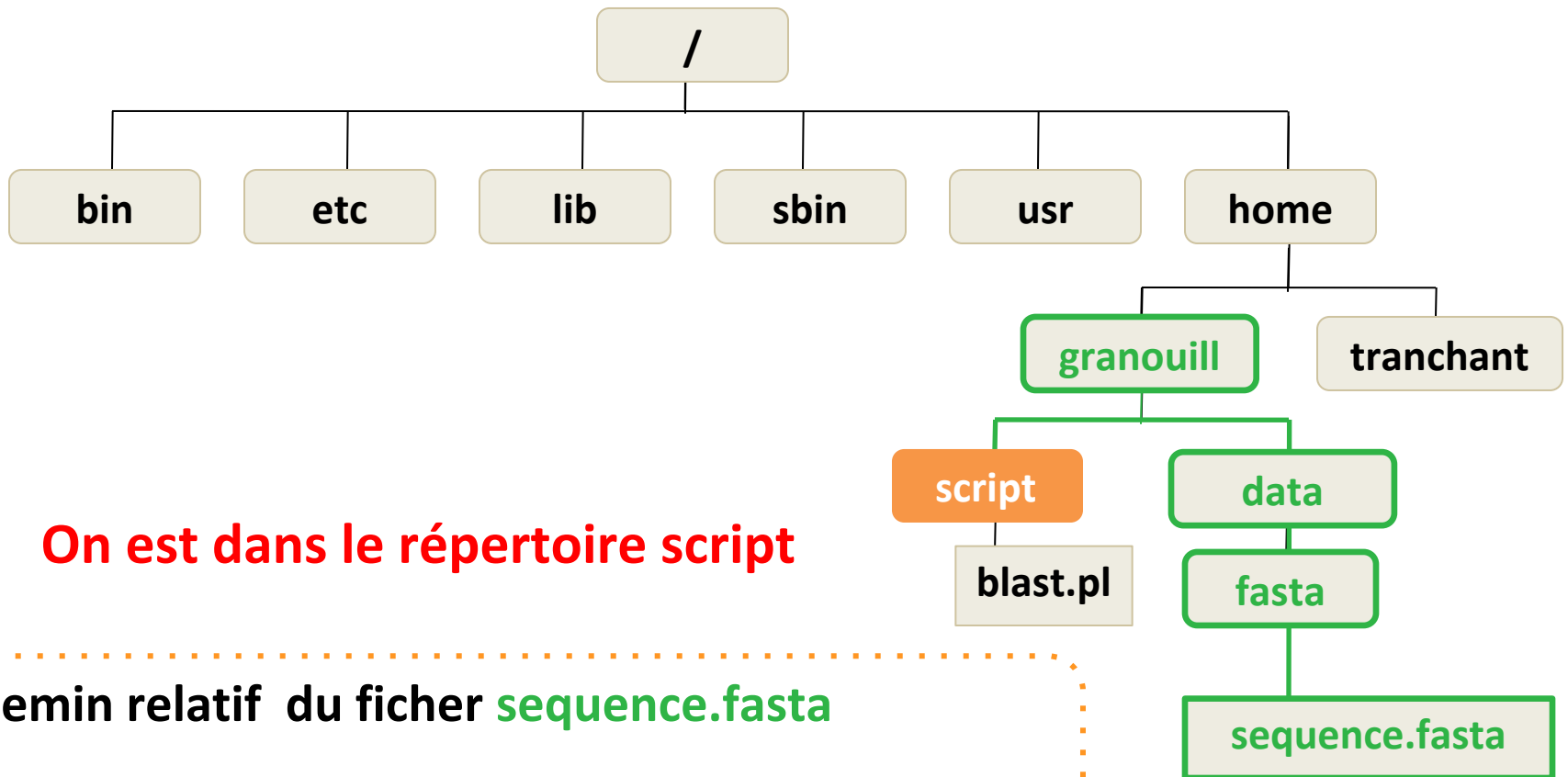
- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



- Défini par rapport où on est dans l'arborescence
- **Ne commence jamais par /**



On est dans le répertoire script

Chemin relatif du fichier **sequence.fasta**

../data/fasta/sequence.fasta



Practice

Is

4

Go to [Practice 4](#) on our github

pour naviguer dans l'historique de vos commandes

<↑> <↓>

pour passer en revue les commandes utilisées avant

<Ctrl> + r

Rechercher une commande précédemment tapée

<Tab>

Complète automatiquement un nom s'il est unique

<Tab><Tab>

Affiche la liste des différentes possibilités si le choix
n'est pas unique

<Ctrl> + c

Tue le processus en cours dans la console

- Linux est sensible à la case
- Utiliser uniquement des lettres, chiffres et caractères _ . –
- PAS d'espaces, accents et caractères spéciaux qui ont des significations particulières pour linux

& ~ # " ' { ([| ` \ ^ @)] } \$ * % ! / ; , ?

- Le suffixe des noms de fichiers (.txt, . fasta, .fa, .fq etc.) est optionnel et peut contenir des chiffres ou des lettres



Commandes relative à l'arborescence de fichiers

Métacaractères : *, []

ls -l
list long

*liste les fichiers avec des informations pour
chaque fichier/répertoire*

On peut utiliser la commande ls sur un jeu de fichiers dont on ne connaît pas les noms en utilisant les caractères spéciaux (Metacharacters)

*

N'importe quelle chaîne de caractère



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

```
ls *fastq
```

*

N'importe quelle chaîne de caractère



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

ls *fastq

KYVF-01.R1.fastq

KYVF-02.R1.fastq

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

*

N'importe quelle chaîne de caractère



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

```
ls KYVF*fastq
```

*

N'importe quelle chaîne de caractère



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

ls KYVF*fastq

KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

*

N'importe quelle chaîne de caractère

[ensemble]

Tous les caractères dans l'ensemble

[!ensemble]

Tous les caractères pas dans l'ensemble



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

```
ls *. [sb]am
```

*

N'importe quelle chaîne de caractère

[ensemble]

Tous les caractères dans l'ensemble

[!ensemble]

Tous les caractères pas dans l'ensemble



KYVF-01.R1.fastq

KYVF-02.R1.fastq

KYVF.sam

ZO16.fastq

KYVF-01.R2.fastq

KYVF-02.R2.fastq

KYVF.bam

ZO16.bam

```
ls *. [sb]am
```

KYVF.sam ZO16.bam

KYVF.bam

```
= ls *. [!f]*
```



Practice

ls, *

5

Go to [Practice 5](#) on our github



Commandes relative à l'arborescence de fichiers

commande `cd`

cd

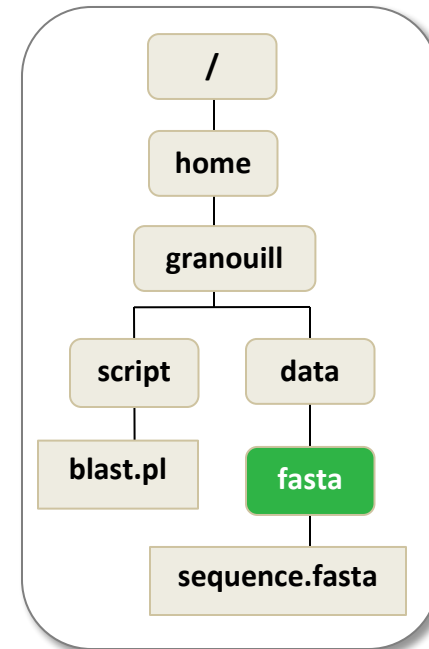
pour se déplacer dans l'arborescence

Change Directory

```
cd nom_repertoire (chemin absolu ou relatif)
```

cd nom_repertoire(chemin absolu ou relatif)

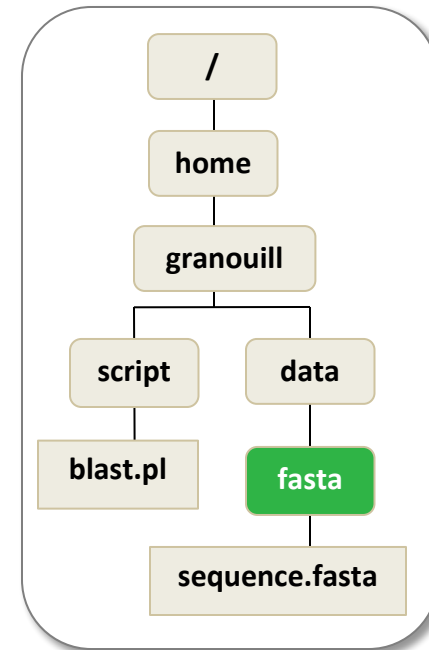
Chemin absolu :



cd nom_repertoire(chemin absolu ou relatif)

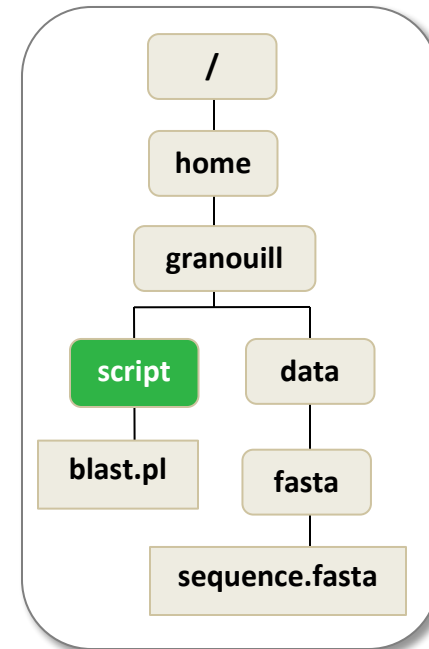
Chemin absolu :

cd /home/granouill/data/fasta



cd nom_repertoire(chemin absolu ou relatif)

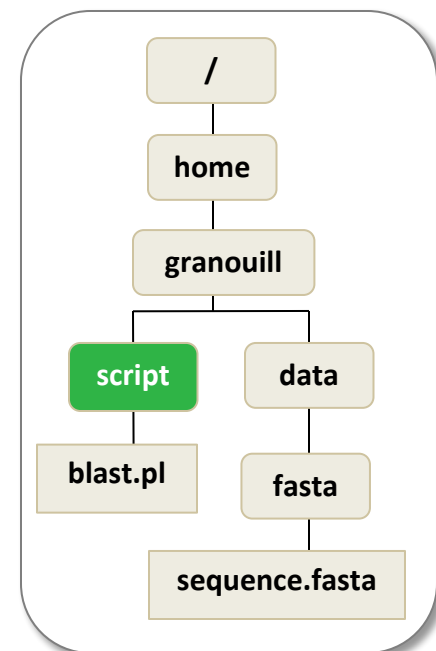
Chemin absolu :



cd nom_repertoire(chemin absolu ou relatif)

Chemin absolu :

cd /home/granouill/script/



`cd nom_repertoire(chemin absolu ou relatif)`

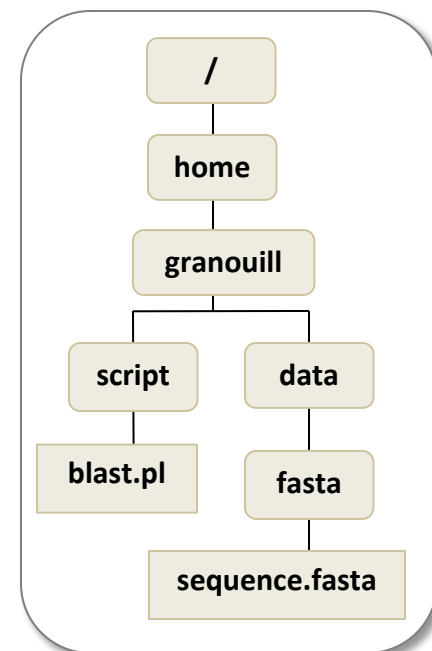
Chemin relatif :

| Commande | Se déplace dans |
|--------------------------------|-----------------------------|
| <code>cd directory_name</code> | <code>directory_name</code> |
| <code>cd</code> | <i>home directory</i> |
| <code>cd ..</code> | le répertoire parent |
| <code>cd ../..</code> | ? |

Retour au répertoire de l'utilisateur courant home

Remonte d'1 répertoire

Remonte de 2 répertoires



Arborescence linux

pwd

Affiche le chemin absolu

ls

Liste tous les fichiers/répertoires

cd rep_name

Se déplace dans rep_name

mkdir rep_name

Crée un répertoire

rm nom_fichier

Supprime un fichier

cp fichier_source repertoire_cible

Copie un fichier dans un répertoire

cp fichier_source nouveau_fichier

Copie un fichier sous un nouveau nom



Practice

`pwd, cd, ls`

6

Go to [Practice 6](#) on our github



Practice

`mkdir, mv, ls, cp, rm`

7

Go to [Practice 7](#) on our github



**Des commandes pour éditer
des fichiers et les manipuler**

cat nom_fichier

Affiche le contenu d'un fichier sans pagination

```
MacBook-Pro-de-Christine:Data tranchan$ cat Data/Fasta/sequence.fasta
>Gxbjbsjxbjs
CCACCCCTCTTACAGTCTTCACCAAATGTCCTTTAAAACTCCACCTAAAGTATCCAAAGA
CTCGAGAAATGCTGTGCCACAACCAGCTTTTGAGTCATCCATGACCGTTGATCTTCCTTT
GCCCCCAGAGTGGGGCCTAGCACCATCTAGCTACTACTTGCCCTTTCATACCCATCATTGG
GATACCCTGAATACCTATCTTATAAGTTCCATATGGCTTATATTTCTAAGTAAGAGATGC
ACTTAGTAAGTGCATGTCGTCTTGACTTGTTTATACTCTAATGTATGATATTTATATCCC
TATAATATAGTGTTACTAATATATGTTTGGTATTGTGTAGACTCCATTGTACCATGGTGT
GCTAATTAGAAATAACATGCCAGCTTTGCTATTGTGGTTTGCAAGTAAAGTAAAAAAA
MacBook-Pro-de-Christine:Data tranchan$
```

Affiche le contenu d'un fichier

<space> pour défiler

q pour quitter

```
less Data/Fasta/EST-68566-Coffeacaneophora.fasta
```

AATATTTTGGTGTCAACGGTTCACCGCTAATCTTTCTGCACAAAGTAAGAGAAGGGTAACAAGTAAAAGGG
 AGCGAGACTATGGCTGCCACATCTGCTGCAGTACTAAATGGATTGGGCTCTCCCTTCTTGACTGGTGGAA
 GGAAAAGTCAGGCCCTGCTGGCTTCACCAGCTGCAGCTAGAGTCGGTGGTGCTGCTGCTGTTGCTCCAAA
 GAGATTAGTTGTGGTAGCTGCTCGTCTCCCAAGAAGTCATGGATTCTGGTGTTAGAACTGGTCCAGGC
 TTCCCTTGACCCCGAGTATCTCGATGGCTCGCTCCCTGGTGACTACGGTTTTGATCCACTTGGTCTTGGCA
 AGGATCCAGCATTCTTGAAATGGTATAGAGAAGCTGAGCTCATTATGGCAGATGGGCAATGACTGCAGT
 GTTGGGAATCTTTGTTGGCCAGGCATGGAGTGGTATCCCATGGTTTCAAGCTGGCGCTGACCCCTGGTGCC
 ATTGCGCCATTCTCCTTTGGTTTCGCTCCTCGGTACTCAACTCCTGCTCA

ATTTCATCGTATCTTGGCCCTCTTTCTTTTCATCTCTCCTCTTCTTTTGATCAGCATCAGAGGCACCGGCAGC
TTTTTAAAAGCTTTATCACACACACACACATTCTTTGTAGTCAGTCAGTGGAACAATTACTGAGTACT
ACTAGTTACTTATCGAACAAATTCGATTAGCTTTGCTAGGAGGTACCGTTTCGGTTGGTTGTACTTGTGTGT
AGTATAAATATTGATGGCTGGGATGCTGTGCGGAGTTAACATTGGAGGCGAGACTGAAACAGCTAAACCA
GTCGAGCCTAATTCTCAGTCGGCTAGGCGGAGGAGGATGGAAATTCATCACTTCATATTTGTAGCTTCGG

Data/Fasta/EST-68566-Coffeacanthophora.fasta

- | | | |
|-------------|---|---|
| head | Affiche les premières lignes d'un fichier (n=10, 10 lignes par défaut) | <code>head -n 20 script.pl</code> |
| tail | affiche les dernières lignes d'un fichier (n=10 par défaut) | <code>tail -n 5 script.pl</code> |
| wc | Compter le nombre de lignes, mots ou caractères d'un fichier | <code>wc script.pl</code> <code>wc -l script.pl</code> |



Practice

Is

8

Go to [Practice 8](#) on our github

grep

pour rechercher un motif dans une ligne

```
grep [options] motif [file1, ... ]
```

grep

pour rechercher un motif dans une ligne

```
grep [options] motif [file1, ... ]
```

| Option | Description |
|--------|---|
| -c | Compte le nombre de lignes dans lesquelles le motif a été trouvées |
| -n | Affiche le numéro de ligne et la ligne dans laquelle le motif a été trouvé |
| -i | Recherche le motif sans tenir compte de la casse |
| -v | Affiche seulement les lignes sans le motif |
| -l | Affiche uniquement les noms de fichiers dans lesquels le motif a été trouvé |



Practice

9

Go to [Practice 9](#) on our github

sort

Trier une ou plusieurs colonnes d'un fichier tabulé

```
sort -k2 fileName
```

sur la colonne 2 (tri alphanumérique)

```
sort -k2r fileName
```

sur la colonne 2, décroissant

```
sort -k2g -k1r fileName
```

sur la colonne 2 et 1

```
sort -t: -k3g fileName
```

sur la colonne 3 (nombre), le délimiteur de
colonne est :

cut

Sélection de colonnes/champs d'un fichier

```
cut -d separateurColonne -f fieldNumber fileName
```

```
cut -d ":" -f1,5 /etc/passwd
```

Sélection des champs 1 et 5 dans un fichier tabulé
dont les colonnes sont séparées par un :



Practice

10

Go to [Practice 10](#) on our github

Formateurs itrop / South Green

- Christine Tranchant-Dubreuil
- Sebastien Ravel
- Alexis Dereeper
- Ndomassi Tando
- François Sabot
- Gautier Sarah
- Bruno Granouillac



Merci pour votre attention !



Le matériel pédagogique utilisé pour ces enseignements est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions (BY-NC-SA) 4.0 International:

<http://creativecommons.org/licenses/by-nc-sa/4.0/>



Autres commandes utiles

Compresser des fichiers **tar, gzip**

```
tar -zcvf tarfile.tar.gz dirToCompress  
gzip fileToCompress
```

Décompresser une archive **gunzip, tar**

```
gunzip file.gzip  
tar -xvf file.tar  
tar -zxvf file.tar.gz
```

Afficher le contenu d'une archive **zcat**

```
zmore data.txt.gz
```

Rechercher une expression/motif dans une archive **zgrep**

```
zgrep 'NM_000020' data.gz
```


rename

| Exemple | Description |
|--|-----------------------------------|
| <code>rename 's/.txt/.fasta/' *</code> | rename the extension of all files |
| <code>rename 'y/a-z/A-Z/' *</code> | rename files in uppercase |

Commande ls -l

```
$ ls -l filename  
drwxrwxrwx 3 user user 4096 2012-02-11 20:21 file_name
```

Permissions

Proprio

Groupe

Taille

Heure et date de la dernière modification

Type

-Interprétation/Légendes des permissions

- : fichier normal

d : répertoire

l : lien symbolique

ls -l command

Permissions

`drwxrwxrwx 3 user user 4096 2012-02-11 20:21 file_name`

other
group
user

3 classes

3 types of permissions :

| Permission | File | Directory |
|-------------|----------------------|---|
| Read r | Ouverture et Lecture | Lister et copier les fichiers |
| Write w | Modiier et supprimer | Manipuler le contenu : copier, créer, modifier, écraser |
| Execution x | Executer le fichier | Accès seulement au fichier pour l'exécuter |

commande pour la gestion des permissions : **chmod**

```
chmod <perm> file_name
```

Chaque permission = 1 valeur

| | |
|------|---|
| R | 4 |
| W | 2 |
| X | 1 |
| none | 0 |

Exemple

```
chmod 740 script.sh
```

```
# Owner=rwx Group=r-- Other=---
```

```
chmod 755 script.sh
```

```
# Owner=rwx Group=r-x Other=r-x
```

chmod, ls

Donner le nom du propriétaire, du groupe et les droits des fichiers contenus dans le répertoire “~/Data/454-projet1/raw”

Changer les droits du fichier Scripts/blast.pl pour qu’il soit :
en lecture/écriture pour le groupe,
lecture/écriture/exécution
pour le propriétaire
et lecture au public

