TD Génomique environnementale

Étude d'un consortium microbien

Version 1.2

5 et 14 mars 2018

Eléonore FROUIN
eleonore.frouin@mio.osupytheas.fr

Table des matières

0 Rappel sur le système Unix

Présentation des commandes Unix essentielles pour naviguer sur un terminal. L'ouverture du terminal se fait via le gestionnaire d'application ou grâce au raccourci **Ctrl+Alt+T**.

0.1 Accéder à une machine distante via ssh

```
$ ssh -X  user@pedaserv2.luminy.univ-amu.fr
```

Pour établir la première connexion ssh, tapez **yes** puis entrez votre mot de passe AMU. L'invite de commande ressemble alors à :

```
user@pedaserv2:~$
```

Pour se déconnecter et revenir sur la machine locale :

```
$ logout
```

0.2 Lister, créer et éditer des fichiers

Liste : La commande ci-dessous permet de lister les fichiers et les sous répertoires présents dans le répertoire courant. À quoi sert l'option **-l** ?

```
$  ls -l
```

Arborescence : Pour naviguer dans l'arborescence des dossiers via un terminal, on utilise la commande **cd** suivi du chemin du dossier cible. Par exemple, pour se placer dans le répertoire **Documents** on utilise :

```
$  cd Documents/
```

Pour remonter vers un dossier parent (dans notre cas, revenir dans le répertoire personnel) :

```
$  cd ../
```

Enfin pour rapidement se placer à la racine de son répertoire personnel, il existe deux solutions :

```
$  cd ~
$  cd
```

ASTUCE : Apprenez à utiliser l'**auto-complétion** avec la touche **Tab** pour compléter le chemin et les noms de fichiers existants sans avoir besoin de les écrire en entier.

Création d'un répertoire La création du nouveau dossier s'effectue le répertoire courant.

```
$  cd Documents
$  mkdir Unix/
$  mkdir Unix/subdir
```

Créez un dossier nommé DIR1 dans le répertoire Unix et déplacez-vous dans ce dossier.

Édition de fichiers À quoi sert la commande ci-dessous ?

```
$  touch fichier.txt
```

L'édition de fichier peut être réalisée dans le terminal via des éditeurs de texte tels que vim, emac, nano ..

```
$  nano fichier.txt
```

Écrivez deux lignes de texte puis quittez l'éditeur en sauvegardant votre fichier.

Pour connaître le contenu d'un fichier, on utilise la commande **less** ou encore la commande **more**.

```
$  less fichier.txt
```

Taper **q** pour quitter

0.3 Copier, déplacer et renommer des fichiers ou dossiers

Copier La commande **cp** permet de copier un fichier vers un autre fichier ou dossier.

```
$  cp fichier.txt fichier_copie1.txt
$  cp fichier.txt ../subdir/fichier_copie2.txt
```

La copie d'un dossier nécessite l'ajout de l'option **-r**.

```
$  cd ~/Documents/Unix/
$  cp -r DIR1/ DIR2/
```

Déplacer La commande **mv** permet de déplacer un ou plusieurs fichiers et/ou dossiers. L'argument final désigne la destination et est obligatoirement un répertoire.

```
$  cd ~/Documents/Unix/DIR1
$  mv fichier.txt fichier_copie1.txt ../subdir/
```

Renommer La commande pour renommer un fichier ou un dossier est également **mv**.

```
$  cd ~/Documents/Unix/DIR2
$  mv fichier.txt Nouveau.txt
```

Déterminez les actions des commandes ci-dessous (renomme et/ou déplace). On supposera que tous les dossiers en majuscule existent, mis à part le dossier NEW_DIR.

```
mv test.txt  data
mv test.txt  DATA/
mv test.txt  ../../seq.fasta ../../TEST/
mv seq.fasta ../../TEST/seq.fasta
mv DATA ../../NEW_DIR
```

0.4 Supprimer des fichiers ou dossiers

La commande **rm** permet de supprimer des fichiers. Elle doit être complétée par l'option -r pour la suppression des dossiers.

```
$  cd ~/Documents/Unix/
$  rm  DIR2/Nouveau.txt
$  rm -r DIR1
```

Attention : La commande **rm** détruit directement les fichiers, sans passage habituel par la "corbeille".

0.5 Quelques dernières commandes : echo, grep

La commande **echo** permet d'afficher une chaîne de caractères sur le terminal (sortie standard).

```
$  echo "j'adore Unix"
$  echo "j'adore Unix"> sortie.txt
```

Que se passe-t-il ? Que contient le fichier *sortie.txt* ?

```
$  echo "je ne sais pas"> sortie.txt
```

Vérifiez à nouveau le contenu du fichier *sortie.txt*.

```
$  echo "si j'adore Unix" >> sortie.txt
```

Que fait le $>>$?

La commande **grep** recherche dans un ou plusieurs fichiers les lignes contenant un certain motif. L'option **--color** permet de colorer le motif recherché sur la sortie standard du terminal.

```
$  grep  --color 'Unix' sortie.txt
$  grep  --color 'i' sortie.txt
```

Expliquez ce que renvoie la commande ci-dessous.

```
$  grep  -c 'i' sortie.txt
```

1 Introduction

1.1 Contexte de l'étude

L'article de Koenig *et al* (cf ANNEXE) présente une étude sur la succession des consortia bactériens de la flore intestinale de nourrissons au cours de leur deux premières années. Les auteurs utilisent ici deux techniques : le méta-barcoding et la métagénomique, pour caractériser les communautés microbiennes. Leur analyse les a conduit à deux conclusions :

1. Globalement la diversité phylogénétique et la composition des communautés évoluent graduellement avec le temps.
2. Certains groupes taxonomiques majoritaires peuvent en revanche présenter des brusques changements d'abondance en fonction du régime alimentaire et de la santé.

Méta-barcoding 16S Le séquençage ADN du gène de l'ARN ribosomique 16S permet de reconstruire l'histoire évolutive des organismes. Ainsi on peut étudier la taxonomie des espèces présentes, réaliser une analyse de la diversité, et également quantifier les proportions des taxons suivant différentes conditions.

Métagénomique Il s'agit du séquençage direct de l'ADN présent dans un échantillon. Dans le cadre de l'article d'étude, il s'agit de matière fécale. La métagénomique consiste à identifier les gènes présents dans la communauté microbienne, et à assigner des fonctions aux gènes identifiés. Cette technique donne un aperçu du potentiel fonctionnel d'un environnement.

1.2 Les données

Les deux types de données, utilisée dans l'article de Koenig *et al*, sont à disposition pour aborder l'analyse :

— Des données d'amplicons du gène de l'ARNr 16S (55 échantillons), disponibles sur MG-RAST
`http://metagenomics.anl.gov/?page=MetagenomeProject&project=65`

— Des données de métagénomes (12 échantillons) disponibles sur EBI Metagenomics `https://www.ebi.ac.uk/metagenomics/projects/SRP002437`

1.3 Fichiers de séquences biologiques

Le format FASTA Format de fichier texte utilisé pour stocker des séquences nucléiques ou protéiques. Une entrée d'un fichier FASTA est constituée de **deux lignes**. La première décrit la séquence en commençant par le signe `">"` suivi de l'identifiant de la séquence. La deuxième ligne contient la séquence en elle-même.

```
> SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
```

L'extension d'un fichier FASTA est conventionnellement *.fasta* ou *.fa*. Il est également possible de trouver les extensions *.fna* et *.faa* pour distinguer les nucléotides des acides aminés.
Le format FASTA se prête facilement à la manipulation et à la lecture des séquences via des outils de traitement de texte et de langages script tels que Perl.

Le format FASTQ Format de fichier texte permettant de stocker à la fois des séquences nucléiques et les scores de qualité associés, établis lors du séquençage. Une entrée d'un fichier FASTQ est constituée de **quatre lignes**. La première commence par un caractère `"@"` suivi de l'identifiant de la séquence. La deuxième ligne contient la séquence nucléique. La troisième commence par un caractère "+", parfois suivi d'une description de la séquence. La dernière ligne

contient les scores de qualité associés à chacune des bases de la séquence de la ligne 2. Le score de qualité est codé par un caractère ASCII.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

L'extension d'un fichier FASTQ est conventionnellement *.fastq* ou *.fq*.
Grâce au score qualité associé à chacune des bases, il est possible de filtrer les séquences, en ne conservant que celles dont la qualité est supérieure à un seuil donné.

Manipulation des fichiers FASTA

ATTENTION : Les fichiers FASTA peuvent être très volumineux, il est donc déconseillé de les ouvrir dans un éditeur de texte (tel que Gedit). Il est préférable d'utiliser les commandes **less** ou **more** pour les afficher dans un terminal.

Trouvez la commande Unix qui permet de compter le nombre de séquences dans un fichier fasta. Testez-la sur le fichier *test.fasta*.

1.4 Liste des outils

Il existe de très nombreux outils bioinformatiques pour réaliser chacune des étapes de l'analyse génomique et il n'est pas toujours facile de s'y retrouver. Le site `omictools.com/` recense les principaux. Ces outils, logiciels, peuvent soit être installés en local, soit lancés depuis des interfaces Web (les calculs sont réalisés sur des serveurs distants). Les interfaces Web sont souvent limitées en terme de taille de jeux de données à traiter, mais ne nécessitent aucune installation et ont une prise en main plus aisée pour des non-spécialistes.

1.5 Principales étapes du traitement de données métagénomiques

1.5.1 Analyse d'amplicons 16S

	Exemple d'outils bioinfo
1. Filtrage qualité des séquences des amplicons	Trimmomatic
2. Suppression des chimères créées pendant le traitement d'amplification PCR	DECIPHER
3. Alignement des séquences entre elles pour pouvoir les comparer	muscle
4. Regroupement des séquences similaires ou très proches en OTU (Unité Taxonomique Opérationnelle) et sélection pour chaque OTU de la séquence la plus représentative	uclust, usearch61
5. Assignation et classification des séquences retenues (au niveau du phylum, du genre, de l'espèce ...)	PhymmBL
6. Analyse, comparaison (sortie graphique ..)	phyloseq (R)

1.5.2 Analyse des métagénomes

Les métagénomes contiennent les informations relatives à la composition taxonomique des communautés microbiennes étudiées. Deux méthodes d'analyse sont envisageables ; leurs principales étapes sont présentées ci-dessous.

A.1. Analyse taxonomique via des gènes spécifiques

1. Filtrage qualité (en fonction du score qualité des fichiers fastq, suppression des réplicats artificiels).
2. Identification de gènes marqueurs de la phylogénie (exemple : gène de l'ARNr 16S, gène mcrA pour les méthanogènes).
3. Affectation de ces gènes à une taxonomie.

Quel est le principal désavantage de cette méthode ?

A.2. Analyse taxonomique basée sur le potentiel protéique

1. Filtrage qualité (fastq.score, suppression des réplicats artificiels)
2. Comparaison de l'ensemble des séquences avec les bases de données protéiques existantes pour déterminer leur taxonomie.

L'analyse des métagénomes permet d'aller bien au-delà de l'analyse taxonomique, en estimant par exemple le potentiel fonctionnel des communautés microbiennes.

B. Analyse fonctionnelle

1. Filtrage qualité (fastq.score, suppression des réplicats artificiels).
2. Assemblage des séquences génomiques en longs fragments.
3. Détection des cadres ouverts de lectures dans les longs fragments d'ADN.
4. Annotation fonctionnelle des gènes identifiés.

De nombreuses autres analyses sont possibles : reconstruction de voies métaboliques, détermination des relations entre les individus d'une communauté, description de l'histoire évolutive, et même la reconstruction partielle de génomes.

Quelles sont les techniques d'analyses des métagénomes mises en œuvre dans l'article de Koenig et al ?

2 Tutoriel QIIME

Arborescence de début de projet

```
QIIME_analysis/
    Analysis/                  # dossier d'analyse
        mapfile.csv            # fichier contenant les métadonnées
        Scripts/               # dossier de scripts de visualisation

    Data16S/                   # dossier contenant les données de l'étude
        AllSeqs.fna            # séquences d'amplicons de 55 échantillons
```

Placez le dossier QIIME_analysis dans votre dossier **Documents** et décompressez l'archive QIIME_analysis.

```
$ tar xvf QIIME_analysis.tar.gz
```

Placez-vous dans le **répertoire Analysis** :
TOUTES LES COMMANDES DOIVENT ÊTRE LANCÉES AU NIVEAU DE CE RÉPERTOIRE.

Pipeline QIIME

Le pipeline QIIME (acronyme pour Quantitative Insights Into Microbial Ecology) permet de réaliser l'analyse de séquences microbiennes du gène de l'ARNr 16S. Ce pipeline contient une multitude de programmes qui permettent notamment du filtrage qualité, des alignements taxonomiques, reconstructions phylogénétiques, analyses de diversité et visualisations graphiques.

Toutes les commandes QIIME s'effectuent depuis le serveur **pedaserv2**. Pour la connexion à partir d'un terminal, il faut suivre la procédure indiquée dans la partie 0.1. Pour chaque commande QIIME, une aide est accessible en ajoutant l'option ***-h***, notamment pour avoir accès à l'ensemble des options.

```
nom_de_la_commande.py -h
```

2.1 Regroupement (=clustering) des séquences en OTUs

Lors de cette première étape toutes les séquences des échantillons sont regroupées en unité taxonomique opérationnelle (OTU, groupe de séquences représentant un certain degré de parenté taxonomique). Ces regroupements sont basés sur la **similarité** entre les séquences.

La ligne de commande ci-dessous permet de regrouper les séquences en OTUs en utilisant un seuil à 97% d'identité entre les séquences pour qu'elles soient regroupées.

```
pick_otus.py -i ../Data16S/AllSeqs.fna -o picked_otus/
```

Quels sont les fichiers produits et que contiennent-ils ? Combien d'OTUs ont été créées ? Comment faut-il modifier la commande pour un clustering avec 98% d'identité ?

Facultatif : **Combien de séquences appartiennent au cluster n° 358 ?**

2.2 Choix de la séquence la plus représentative par OTU

Chaque OTU est constituée de plusieurs séquences apparentées, l'étape suivante est de sélectionner **une** séquence représentative par cluster. Cette séquence sera utilisée pour l'identification

taxonomique de l'OTU et l'alignement phylogénétique.

```
pick_rep_set.py -i picked_otus/AllSeqs_otus.txt -f ../Data16S/AllSeqs.fna -o Seqs_rep_set.fna
```

Quel est le format du fichier produit ? Combien contient-il de séquences ?

2.3 Assignation taxonomique des séquences représentatives

Pour affecter une taxonomie aux séquences représentatives choisies, on utilise la méthode uclust consensus taxonomy classifier.

```
assign_taxonomy.py -i Seqs_rep_set.fna -r /usr/local/lib/python2.7/dist-packages/qiime_default_reference/gg_13_8_otus/rep_set/97_otus.fasta
```

Donner un exemple de séquence ayant une affiliation précise au niveau de l'espèce.

2.4 Alignement des séquences représentatives

Les séquences représentatives des OTUs doivent ensuite être alignées entre elles pour permettre l'analyse phylogénétique. Il est ici réalisé avec PyNAST : en combinant notre jeu de séquences et une base de données de gènes d'ARNr 16S (dans le fichier *85_otus.pynast.fasta*), PyNAST réalise un alignement multiple de l'ensemble de ces séquences.

```
align_seqs.py -i Seqs_rep_set.fna -o Pynast_align -t /usr/local/lib/python2.7/dist-packages/qiime_default_reference/gg_13_8_otus/rep_set_aligned/85_otus.pynast.fasta
```

À quoi correspondent les '-' dans le fichier *Seqs_rep_set_aligned.fasta* ?

Combien de séquences n'ont pas pu être alignées ? Comment pourrait-on diminuer ce nombre ?

2.5 Filtrage de l'alignement

```
filter_alignment.py -i Pynast_align/Seqs_rep_set_aligned.fasta -o filtered_alignment/
```

À votre avis, à quoi sert cette étape de filtrage d'après la comparaison des fichiers Seqs_rep_set_aligned.fasta et Seqs_rep_set_aligned_pfiltered.fasta ?

2.6 Construction d'un arbre phylogénétique

Dans cette étape on construit un arbre phylogénétique comprenant les séquences représentatives de chaque OTUs. On le visualisera plus tard dans le TP.

```
make_phylogeny.py -i filtered_alignment/Seqs_rep_set_aligned_pfiltered.fasta -o rep_phylo.tre
```

2.7 Création d'une table d'OTU

La dernière étape consiste à associer les assignations taxonomiques (cf étape 3) avec les regroupements en OTUs (établis lors de la première étape) pour construire un tableau d'abondance d'OTUs.

```
make_otu_table.py -i picked_otus/AllSeqs_otus.txt -t uclust_assigned_taxonomy/Seqs_rep_set_tax_assignments.txt -o otu_table.biom -m mapfile.csv
```

Le format **.biom** stocke les résultats (qui peuvent être volumineux) de façon compressée. Pour pouvoir lire et analyser ces résultats, il faut convertir le fichier dans un format de fichier texte avec séparateur.

```
biom convert -i otu_table.biom  --to-tsv -o otu_table.tsv --header-key taxonomy
```

Comment la table des OTUs est elle construite ? (entête de ligne, de colonne ...) ?

Un résumé de la table d'OTUs peut être obtenu directement (sans convertir le fichier .biom) avec la commande **biom summarize-table**. On peut notamment retrouver le nombre d'OTUs total et par échantillon.

```
biom summarize-table -i otu_table.biom
```

Enregistrez le résultat de cette commande dans un fichier, que l'on nommera *summary_table_otu.txt* .

2.8 Filtrage de la table d'OTUs

Un post-traitement souvent utile consiste à filtrer certaines OTUs présentes dans la table finale.

Regardez la première ligne du fichier otu_table.tsv (après l'entête). Qu'en concluez-vous ?

Par exemple pour ne conserver que les OTUs qui contiennent au minimum 2 séquences, on filtre la table OTUs à l'aide du script filter_otus_from_otu_table.py.

```
filter_otus_from_otu_table.py -i otu_table.biom -n 2 -o otu_table_filtered.biom
```

```
biom convert -i otu_table_filtered.biom  --to-tsv -o otu_table_filtered.tsv
```

```
biom convert -i otu_table_filtered.biom --to-tsv -o otu_table_filtered_taxo.tsv --header-key taxonomy
```

Combien d'OTUs ont été retenus après filtrage ?

***Facultatif :* Quelle est l'option qui filtre la table d'OTUs pour garantir que les séquences de chaque OTU proviennent d'au minimum 2 échantillons différents ?**

2.9 Analyse et représentation graphique

Il est possible de grouper les OTUs à différents niveaux taxonomiques (phylum, classe, ordre, famille, etc.) grâce au script summarize_taxa_through_plots.py.

```
summarize_taxa_through_plots.py -i otu_table_filtered.biom -o taxa_summary -m mapfile.csv
```

À quoi correspondent les différentes tables *.txt* créées dans le dossier taxa_summary ?

3 Analyse avec Rstudio

```
biom convert -i otu_table_filtered.biom  - -to-json -o otu_table_filtered.json
```

3.1 Script graphes.R

Ce script permet de réaliser des graphes simples à partir des fichiers générés par QIIME.
→ Ouvrir Rstudio
→ Ouvrir le script graphes.R
→ Chaque ligne du script peut être executée grâce au raccourci Ctrl+Entrée
→ Si certaines libraries ne sont pas installées dans votre version de R, il faut les importer via Tools-> Install Packages ...
→ Certaines commandes, signalées par # @@ à compléter/modifier @@, doivent être modifiées avant d'être executées ☺

3.2 Script phyloseq.R

Ce script utilise le package phyloseq. Son installation diffère des libraries utilisées jusqu'à présent : il faut copier les lignes ci-dessous dans le terminal R. L'installation est assez lente (environ 15 minutes).

```
source("http ://bioconductor.org/biocLite.R")
biocLite("phyloseq")
```

Le package phyloseq permet le filtrage, le sous-échantilonnage et la comparaison de tables OTUs mais surtout la création de graphiques élaborés.

4 Visualisation avec Krona

Krona est un outil de visualisation interactive. Il génère des diagrammes circulaires à plusieurs niveaux offrant une représentation hiérarchique de la classification taxonomique. Cette visualisation permet notamment d'explorer les abondances d'OTUs au sein d'un taxon spécifique en zoomant sur la section d'intérêt. Ces graphiques sont construits à partir d'une table d'OTUs.

```
biom convert -i otu_table_filtered.biom --to-tsv -o otu_table_filtered_taxo2.tsv --header-key  taxonomy
```

```
Scripts/app_otu_to_krona.pl -i otu_table_filtered_taxo2.tsv -o krona_visualization.html
```

La commande ci-dessus génère un fichier html qui peut être ouvert avec un navigateur web, tel que Firefox.

```
firefox krona_visualization.html
```

Quel pourcentage de *Firmicutes* détecte-t-on dans le microbiote aux jours 4, 58 et 172 ?