Travaux Pratiques de Métagénomique

Comparaison des métagénomes intestinaux de patients en rémission d'infection chroniques de l'intestin et d'individus sains

Bérénice Batut
berenice.batut@udamail.fr

Mars 2015

1 Introduction

[Qin *et al.*, 2010] ont étudié les échantillons fécaux de 124 européens afin de définir et décrire le contenu "minimal" du microbiote de l'intestin humain, à la fois en terme d'organismes et de fonctions présents. Cette étude fait partie du projet européen MetaHit, dont l'objectif principal était d'établir les associations entre les gènes du microbiote intestinale et notre santé et nos maladies. Le jeu de données utilisé dans l'article de [Qin *et al.*, 2010] repose sur des données métagénomiques d'individus sains mais aussi d'individus en rémission à la suite d'infections chroniques de l'intestin.

L'objectif de ce TP est de comparer les métagénomes d'individus sains et d'individus en rémission, principalement en terme d'organismes présents, à partir des données issues du séquençage, récupérées sur la base de données ENA (European Nucleotide Archive) au sein du projet ERP000108.

Chaque groupe est assigné à un jeu de données comprenant un individu sain, nommé `Contrôle`, et un individu en rémission, nommé `Malade`. Chaque membre du groupe doit analyser un métagénome (Contrôle ou Malade) en suivant le protocole décrit dans les sections 2 et 3. Les scripts utilisés doivent être identiques pour tous les métagénomes analysés. Les contenus des métagénomes sont ensuite comparés en suivant les instructions de la section 4.

Les jeux de données bruts sont importants (fichiers de plus d'1 Gb contenant plusieurs dizaines de milliers de reads). Les scripts des sections 2.1 à 3.1 sont seulement testés sur des jeux de données de test disponibles sur l'ENT. A partir de la section 3.1, les analyses et traitements sont effectués sur les vrais jeux de données auxquels chaque groupe est assignés. Ces jeux de données ont été préalablement traités selon le protocole décrit dans les sections 2.1 à 3.1.

Pour ce TP, l'arborescence suivante doit être créée :

```
Groupe<nom du groupe>
├── Scripts
├── Donnees
│   ├── Tests
│   │   └── Graphiques_qualite
│   ├── Malade
│   │   ├── Graphiques_assemblage
│   │   ├── Graphiques_prediction_ORF
│   │   └── Graphiques_affiliation_taxonomique
│   ├── Controle
│   │   ├── Graphiques_assemblage
│   │   ├── Graphiques_prediction_ORF
│   │   └── Graphiques_affiliation_taxonomique
│   └── Comparaison_controle_malade
│       └── Graphiques_affiliation_taxonomique
└── Compte_rendu
```

Des scripts `R`, utiles pour générer les graphiques, sont disponibles sur l'ENT. Ils doivent être

enregistrés dans le répertoire `Scripts`

Un compte-rendu des différents analyses et traitements des métagénomes (sections 2, 3 et 4) est demandé par groupe (comprenant les réponses aux différentes questions posées dans le protocole et les différentes images générées). Ce document et les scripts utilisés serviront d'évaluation pour ce module. Il est demandé de commenter les codes des scripts utilisés. L'ensemble devra être envoyé par mail à l'adresse `berenice.batut@udamail.fr` sous la forme d'une archive, nommée `Groupe<nom du groupe>`, contenant le compte-rendu au format `PDF` et le dossier `Scripts` avec les différents scripts développés, avant le 31 Mars 2015 23h59.

2 Prétraitements des séquences

2.1 Gestion des fichiers et du type de données

Les fichiers contiennent les séquences des métagénomes (WGS, "Whole-Genome Sequencing"), obtenues par séquençage Illumina (modèle Illumina Genome Analyzer II) en "paired-end".

Question 1 Pour chaque métagénome, deux fichiers sont fournis. A quoi correspondent chacun des fichiers ?

Tâche Réaliser un script `Perl/BioPerl`, nommé `gestion_fichiers.pl`, qui :

- Transforme chaque fichier `fastq` en un fichier `fasta` et un fichier `qual` et calcule la taille des séquences dans un fichier `texte`. Faire une méthode `transforme_fastq_en_fasta_qual` qui :
  - Prend en argument le chemin vers le fichier `fastq` d'entrée, les chemins vers les fichiers `fasta`, `qual` et `texte` qui contiendra les longueurs des séquences avec une séquence par ligne
  - Parcoure les séquences du fichier d'entrée et qui, pour chaque séquence (si elle n'est pas vide) :
    - Écrit la séquence dans le fichier `fasta` de sortie
    - Écrit la séquence des scores dans le fichier `qual` de sortie
    - Récupère la taille de la séquence et l'enregistre dans le fichier `texte` (avec l'identifiant de la séquence, séparé par une tabulation
  - Ferme le fichier `texte`

Question 2 Pourquoi le fichier `qual` est-il plus lourd que le fichier `fastq` ?

Tâche Compléter le script `gestion_fichiers.pl` pour qu'il :

- Applique le script `calcule_statistiques_longueur_sequences.R` via la commande `Rscript Scripts/calcule_statistiques_longueur_sequences.R <chemin vers le fichier contenant les longueurs>` pour les deux fichiers

Question 3 Combien y a-t-il de séquences initialement dans les fichiers ? Quelles sont la taille moyenne, la taille maximale et la taille minimale des reads ?

Tâche Compléter le script `gestion_fichiers.pl` pour qu'il :

- Compare les identifiants conservés pour les deux fichiers afin de conserver l'ensemble des identifiants présents dans les deux fichiers. Pour cela, faire une méthode `retrouve_id_a_conserver` qui :
  1. Prend en argument les chemins vers les fichiers contenant les longueurs des séquences
  2. Applique la commande UNIX `diff` pour comparer le contenu des deux fichiers, en récupérant la sortie de la commande dans un fichier `sortie_diff`
  3. Parcoure le contenu du fichier `sortie_diff` pour déterminer les identifiants correspondant aux lignes différentes entre les deux fichiers, les enregistre dans un tableau et compte combien de lignes sont modifiés dans chacun des fichiers
  4. Parcoure le contenu du premier fichier passé en argument du `diff` et enregistre dans un tableau les identifiants qui ne posent pas de problème (ceux qui n'ont pas été identifiés

au point précédent). Pour tester si `element` est dans le tableau `list`, il faut utiliser la commande `$element ~~ @list`

5. Renvoie, en référence, le tableau contenant les identifiants à conserver ainsi que le nombre de lignes à éliminer des deux fichiers

— Imprime à l'écran la taille du tableau renvoyé (nombre de séquences à conserver) ainsi que le nombre de séquences à éliminer dans chacun des deux fichiers

Question 4 Combien de séquences doivent être éliminées ?

Tâche Compléter le script `gestion_fichiers.pl` pour qu'il :

— Élimine les séquences qui ne sont pas présentes dans les deux fichiers dans les fichiers `fasta` et `qual` (ne l'appliquer qu'aux fichiers de test). Faire une méthode `conserve_sequences` qui :

— Prend en argument le chemin vers un fichier `fasta` ou `qual`, le type de fichier (`fasta` ou `qual`), le chemin vers le fichier de sortie et le tableau des identifiants à conserver dans ce fichier

— Parcoure le contenu du fichier d'entrée et pour chaque séquence, tester si l'identifiant est dans la liste des identifiants à conserver et si oui, l'enregistre dans le fichier de sortie les séquences. Pour tester la présence de l'identifiant dans la liste des identifiants à conserver, implémenter la fonction `teste_id_dans_liste_ids` correspondant au pseudo-code suivant :

Données:
`id` : chaîne de caractères
`liste` : tableau de `N` entiers, correspondant à la liste des identifiants à conserver transformés en entier
Sorties: booléen
Transformation d'`id` en un entier;
`continue` = vrai;
`i` = 0;
tant que *continue* = *vrai* **faire**
  si *i* = *N* **alors**
    `continue` = faux ;
  sinon si *liste[i]* >= *id* **alors**
    `continue` = faux;
  sinon
    i = i + 1;
  fin
fin
`trouve` = faux;
si *i* < *N* **alors**
  si *liste[i]* = *id* **alors**
    `trouve` = vrai;
  fin
fin

Question 5 En quoi le pseudo-code fourni est-il plus efficace qu'un test standard de présence d'un élément dans une liste ?

Question 6 Pourquoi cette étape de traitement des fichiers doit-elle être effectuée ? Quels sont les risques si elle n'est pas faite ?

2.2 Suppression des primers, adaptateurs et tags d'indentification

Question 7 A quoi correspondent les primers, adaptateurs et tags d'identification dans le cadre du séquençage Illumina ?

Question 8 En explorant les fichiers `fasta` à l'aide des commandes `more` ou `less`, y a-t-il des similarités dans les extrémités des séquences ? Que faut-il faire si c'est le cas ? Expliquer brièvement la démarche à mettre en place.

2.3 Contrôle de la qualité des séquences et des biais de séquençage

Question 9 Quel est le score de qualité limite pour que la probabilité d'erreur d'identification ne soit pas supérieure à 0.05 ? Détailler la démarche suivie pour obtenir ce résultat.

La qualité des séquences décroit avec la longueur des séquences. Pour déterminer la taille minimale des séquences à conserver, il faut faire un graphique représentant les scores de qualité le long des séquences. Pour cela, le script `genere_graphique_score_qualite.R` est utilisé. Cependant, ce script a besoin d'un tableau où les colonnes sont les positions le long des séquences et les lignes les différentes séquences ; dans une cellule située à la colonne y et à la ligne x, il y a le score de qualité de la séquence x à la position y. Le nombre de colonne du tableau est la taille maximale des séquences.

Tâche Réaliser un script `Perl/BioPerl`, nommé `controle_qualite.pl`, qui :
- Transforme les fichiers `qual` en un fichier contenant le tableau demandé. Faire une méthode `transforme_qual_en_tableau` qui :
  - Prend en argument le chemin vers le fichier `qual` à transformer et le chemin vers le fichier `texte` de sortie
  - Parcoure les séquences du fichier `qual` et, pour chaque séquence :
    - Concatène la séquence des scores de qualité en séparant par des espaces
    - Ajoute le nombre d'espaces nécessaires pour remplir toutes les colonnes (c'est-à-dire atteindre la taille maximale des séquences)
    - Écrit les informations dans le fichier de sortie
  - Ferme le fichier `texte`
- Applique le script `genere_graphique_score_qualite.R` via la commande `Rscript Scripts/genere_graphique_score_qualite.R <chemin vers le fichier avec le tableau de qualités> <chemin vers l'image de sortie en pdf dans le répertoire Graphique_qualite>`

Question 10 D'après les graphiques générés, quelle est la taille limite des séquences tel que le score de qualité reste correct ? Pourquoi ?

Tâche Compléter le script `controle_qualite.pl` pour qu'il coupe des séquences au-delà du seuil de longueur défini. Faire une méthode `filtre_sequences` qui :
- Prend en argument le chemin vers le fichier `fasta` à transformer, le chemin vers le fichier `fasta` de sortie et le seuil défini
- Parcoure les séquences du fichier `fasta` d'entrée et, pour chaque séquence :
  - Coupe la séquence au seuil passé en argument
  - Enregistre la séquence dans le fichier de sortie

Les nouvelles méthodes de séquençage peuvent entraîner des biais, en particulier des biais dans les fréquences des différentes bases surtout en début de séquence. Il faut vérifier ces biais en regardant les fréquences de différentes bases (A, T, C et G) le long des séquences.

Tâche Compléter le script `controle_qualite.pl` pour qu'il :
- Détermine les fréquences de nucléotides le long des séquences et les enregistre dans un fichier `texte` où les colonnes sont les différentes positions le long des séquences et les lignes les fréquences des différents nucléotides moyennées sur l'ensemble des séquences d'un fichier `fasta`. Faire une méthode `determine_frequence_nucleotide` qui :
  - Prend en argument le chemin vers le fichier `fasta` avec les séquences, le chemin vers le fichier `texte` de sortie et la taille maximale des séquences correspondant au seuil défini précédemment
  - Initialise un tableau d'association où les clés sont les différents nucléotides possibles (A, T, C, G et N) et les valeurs des tableaux de taille correspondant à la taille maximale des séquences et initialisés seulement avec des 0
  - Initialise un compteur pour compter le nombre de séquences
  - Parcoure les séquences du fichier `fasta` d'entrée et, pour chaque séquence :
    - Récupère la séquence
    - Parcoure la séquence et, pour chaque base, ajoute un à la position correspondante dans le tableau correspondant au nucléotide rencontré dans le tableau d'association
    - Incrémente le compteur de séquence

  - Parcoure les tableaux de comptage des différents nucléotides (dans l'ordre A, T, C, G et N) dans le tableau d'association et pour chaque valeur, l'enregistre dans le fichier de sortie en prenant soin de diviser chaque valeur par la valeur du compteur (pour avoir des proportions d'utilisation de chacune des bases aux différentes positions)
  - Ferme le fichier `texte` de sortie
- Exécute le script `genere_graphique_frequences_nucleotides.R` via la commande `Rscript Scripts/genere_graphique_frequences_nucleotides.R <chemin vers le fichier avec le tableau de fréquences> <chemin vers l'image de sortie en pdf dans le répertoire Graphique_qualite>`

Question 11 Existe-t-il un biais dans les fréquences des différents nucléotides ? Si oui, que faire pour y remédier ?

3 Traitement des données métagénomiques

3.1 Assemblage des données

L'assemblage des données se fait avec le logiciel `SOAPdenovo2`, sur les fichiers `fastq` bruts en utilisant les informations collectées dans la section précédente. Le logiciel `SOAPdenovo2` permet l'assemblage de données Illumina en utilisant des *k-mers*[1] et des graphes *de Bruijn*[2].

Question 12 Quelle est la taille idéale pour les `k-mers`, sachant qu'elle doit correspondre au nombre impair le plus proche supérieur ou égal à la moitié de la taille moyenne des reads, après coupure ?

Les fichiers de test ne contiennent pas assez de séquences pour obtenir des assemblages corrects. Le suite du travail se fait donc sur les vrais données. Cependant, l'assemblage *de novo* est une étape longue et coûteuse en terme de ressource (principalement en mémoire vive). Ainsi, l'assemblage des données a été effectuées en amont des TPs. Pour continuer, il faut donc demander les sorties de l'assembleur correspondant au jeu de données à étudier.

Question 13 D'après les fichiers sortie de l'assembleur, quelles sont les statistiques d'assemblage ?
- Nombre de contigs/scaffolds
- Taille moyenne/médiane/maximale/minimale des contigs/scaffolds
- N50 des contigs/scaffolds
- Pourcentage de reads dans des contigs
- Pourcentage de contigs dans des scaffolds
- Nombre moyen de contigs par scaffolds
- Nombre de N dans les contigs/scaffolds

Comparer les statistiques entre les deux individus.

Question 14 La suite des analyses se fait en utilisant les contigs plutôt que les scaffolds. Pourquoi ce choix est-il fait ?

Tâche Compléter le script `assemblage.pl` pour qu'il :
- Récupère les longueurs et la couverture par les *k-mers*[3] des contigs générés en parcourant les fichiers `contigs.fasta` contenant les séquences et en enregistrant l'identifiant, la longueur et la couverture de chacune des séquences dans un fichier `texte` avec une séquence par ligne. Faire une méthode `retrouve_longueur_sequences`.
- Exécute le script `genere_graphiques_longueur_contig_scaffold.R` via la commande `Rscript Scripts/genere _graphiques_contig.R <chemin vers le fichier avec les informations des contigs> <chemin vers le répertoire Graphiques_assemblage>`

1. Les *k-mers* sont toutes les sous-chaînes possibles de longueur k contenue dans une chaîne de caractères.
2. Un graphe de de Bruijn est un graphe orienté qui permet de représenter les chevauchements de longueur $n-1$ entre tous les mots de longueur n sur un alphabet donné. Dans le cadre de l'assemblage de génomes, les nœuds sont toutes les sous-séquences de taille k, dits *k-mers*[1], apparaissant dans les reads et les arrêtes correspondent aux sous-chaînes de longueur $k+1$.
3. La couverture est dans le descriptif des séquences (entre le mot `cvg` et `tip`)

Question 15 En analysant les graphiques générés, expliquer pourquoi il faut éliminer les contigs de taille inférieure à 100 bp ?

Tâche Compléter le script `assemblage.pl` pour qu'il :
- Élimine les contigs de taille inférieure à 100 bp et de couverture inférieure à 5 à l'aide d'une méthode `filtre_contigs`. Cette méthode prend en argument les seuils de taille et de couverture acceptés et les noms des fichiers d'entrée et de sortie

Question 16 Combien de contigs sont conservés ?

3.2 Prédiction des ORF

Pour l'affiliation fonctionnelle et taxonomique, il faut comparer les séquences aux séquences disponibles dans les bases de données, c'est-à-dire à des gènes. Il est ainsi nécessaire de rechercher des gènes dans les séquences. Pour cela, des outils de prédiction des ORF comme `MetaGeneAnnotator` sont utilisés.

Tâche Réaliser un script `Perl/BioPerl`, nommé `prediction_orf.pl`, qui :
- Exécute `MetaGeneAnnotator` avec la commande `/usr/local/bin/mga_linux_ia64 -m <chemin vers le fichier contenant les contigs filtrées> > <chemin vers un fichier prediction _orf_sortie>`

Le fichier de sortie se présente de la façon suivante :

```
# [sequence name]
# gc = [gc%], rbs = [rbs%]
# self: [(b)acteria, (a)rchea, (p)hage,(u)nused]
[gene id] [start pos] [end pos] [strand] [frame] [complete/partial] [gene score] [used model] [rbs start] [rbs end] [rbs score]
[gene id] [start pos] [end pos] [strand] [frame] [complete/partial] [gene score] [used model] [rbs start] [rbs end] [rbs score]
[gene id] [start pos] [end pos] [strand] [frame] [complete/partial] [gene score] [used model] [rbs start] [rbs end] [rbs score
# [sequence name]
# gc = [gc%], rbs = [rbs%]
# self: [(b)acteria, (a)rchea, (p)hage,(u)nused]
[gene Id] [start pos] [end pos] [strand] [frame] [complete/partial] [gene score] [used model] [rbs start] [rbs end] [rbs score]
[gene Id] [start pos] [end pos] [strand] [frame] [complete/partial] [gene score] [used model] [rbs start] [rbs end] [rbs score]
....
```

11 : contains both start and stop codons
01 : lacks start codon
10 : lacks stop codon
00 : lacks both start and stop codons

FIGURE 1 – Sortie du logiciel `MetaGeneAnnotator`

Tâche Créer une classe `Gene` (constructeur, getters et setters) qui a comme attributs :
- GENE_ID : il doit correspondre à `<nom de séquence>_<identifiant de gène>`
- START_POS
- END_POS
- STRAND
- COMPLETE_OR_PARTIAL
  - **1** complet
  - **2** absence du codon `start`
  - **3** absence du codon `stop`
  - **4** absence des codons `start` et `stop`
- SCORE

Tâche Ajouter à la classe `Gene`
- Une méthode `retourne_informations` qui renvoie une chaine de caractère contenant la taille, le score et l'état, séparés par des tabulations
- Une méthode `teste_longueur` qui prend en argument une longueur et renvoie vrai si la séquence est plus longue que la longueur passée en argument et faux sinon

— Une méthode `crée_Bio_Seq_proteine` qui renvoie un objet `Bio::Seq`. Cette méthode prend en argument l'objet `Bio::Seq` contenant la séquence du contig pour lequel l'ORF a été prédit. L'objet `Bio::Seq` créé et renvoyé contient avec la séquence correspondant à l'ORF prédit (entre `START_POS` et `END_POS`), l'identifiant de la séquence et il traduit en protéine

Tâche Compléter le script nommé `prediction_orf.pl` pour qu'il :
— Retrouve les informations sur les gènes prédit dans le fichier de sortie et les rentre dans un tableau d'association où les clés sont les noms des séquences et les valeurs des références sur des tableaux contenant des instances de la classe `Gene` créées avec les gènes prédits pour chaque séquence. Faire une méthode `retrouve_orf` qui :
  — Parcoure le fichier
  — Enregistre pour chaque séquence les gènes prédits dans le tableau d'instances de la classe `Gene`
  — Enregistre dans un fichier `texte` pour chaque gène prédit sa taille, son score et son état en utilisant la méthode `retourne_informations`
  — Renvoie le tableau d'association par référence
— Exécute le script `genere_graphiques_lg_score_orf.R` via la commande `Rscript genere_graphiques_lg_score_orf.R <chemin vers le fichier avec les informations des ORF> <chemin vers le répertoire Graphique_prediction_orf>`

Question 17 Combien d'ORF sont prédits ? Quelle est leur taille moyenne, leur taille médiane, leur taille minimale et leur taille maximale ? Comment sont répartis les ORF entre les différentes états (complets, absence du codon `start`, ...) ?

Question 18 D'après les graphiques générés, pourquoi conserver seulement les séquences de taille supérieure à 100 paires de bases ? Pourquoi ne pas éliminer les séquences non complètes ?

Tâche Compléter le script nommé `prediction_orf.pl` pour qu'il :
— Extrait les séquences des gènes prédits de taille supérieure à 100 paires de bases dans le fichier contenant les séquences des contigs. Faire une méthode `extrait_sequences_orf` qui :
  — Prend en argument le chemin vers le fichier `fasta` contenant les contigs filtrés, la référence vers le tableau d'association créé précédemment et le chemin vers le fichier `fasta` de sortie qui contiendra les séquences des gènes prédits, traduits en protéines
  — Parcoure le fichier de séquences et, pour chaque séquence :
    — Parcoure les gènes prédits contenus dans le tableau d'instances `Gene` de cette séquence
    — Détermine les prédits (entre `start_pos` et `end_pos`) dont la taille est supérieure à 100 paires de bases
    — Enregistre les gènes dont la taille est supérieure à 100 paires de bases dans le fichier de sortie en utilisant l'objet `Bio::Seq` renvoyé par la méthode `crée_Bio_Seq_proteine`
  — Compte le nombre d'ORF conservés

Question 19 Combien d'ORF prédits sont conservés ? A quelle proportion des ORF prédits cela correspond ?

3.3 Affiliation taxonomique

Pour l'affiliation taxonomique, c'est-à-dire déterminer les organismes présents dans le métagénome séquencé, nous utilisons le logiciel `MEGAN`. Avant d'utiliser `MEGAN` pour l'affiliation taxonomique des séquences, les séquences du métagénome doivent être comparées à une base de données de séquences de références par une recherche de similarité (`Blast`). Ici, est utilisée une base de données des séquences protéiques de 89 organismes fréquemment trouvés, identifiés par [Qin *et al.*, 2010], dans le microbiote intestinal humain. Pour chaque jeu de données, une base de donnée a été construite, qu'il faut donc demander et enregistrer dans le répertoire `Données`.

Tâche Réaliser un script `Perl/BioPerl`, nommé `affiliation_taxonomique.pl`, qui :
— Transforme le fichier `fasta` contenant les séquences de la base de données en base de données `Blast`

— Effectue une recherche `Blast` des ORF prédits contre la base de données des séquences des 89 organismes fréquemment trouvés, en conservant les hits avec une `e-value` inférieure à 10^{-6}. Faire la recherche de similarité de l'ensemble des séquences contre la base de données en une seule commande
— Parcourir le fichier de sortie `Blast` pour compter le nombre d'ORF qui sont identifiés et le nombre moyen de hit identifiés par ORF

Question 20 Combien d'ORF conservés sont identifiés ? A quelle proportion des ORF conservés cela correspond ? Par combien de séquences en moyenne sont-ils identifiés ?

Tâche Analyser les sorties `Blast` avec `MEGAN` en suivant le protocole suivant :
— Installer `MEGAN` en exécutant `/usr/local/bin/MEGAN_unix_5_9_0.sh`
— Ouvrir `MEGAN`
— Importer la licence `MEGAN` (à demander)
— Importer depuis `Blast` en indiquant le chemin vers le fichier de sortie de l'analyse `Blast` et le fichier avec les ORFs conservés
— Explorer l'arbre phylogénétique des organismes à différents niveaux taxonomiques (domaines, phylum, classes)
— Générer des graphiques des reads assignés aux différents niveaux taxonomiques

Question 21 A quels domaines appartiennent principalement les séquences identifiées ?

Question 22 Parmi les phyla connus (plus de 50), les microbiotes intestinaux humains sont principalement composés de 4 phyla. Lesquels d'après les observations ?

Question 23 Quels sont les biais de cette méthode d'affiliation taxonomiques (utilisation de l'e-value, choix de la base de données, proportion de séquences identifiées, représentativité des séquences identifiées par rapport à l'échantillon complet, ...) ? Que faudrait-il faire pour y remédier ?

3.4 Affiliation fonctionnelle

L'affiliation fonctionnelle suit le même principe que l'affiliation taxonomique : recherche de similarité entre les séquences caractéristiques et des bases de données. Cependant, les bases de données utilisables sont lourdes et sont moins facilement réductibles que celles pour l'affiliation taxonomique. Ainsi, l'affiliation fonctionnelle des séquences n'est pas effectuée ici.

Question 24 Pour identifier les principales catégories des séquences présentes dans l'échantillon, quelle(s) base(s) de données faut-il utiliser ?

Question 25 Pour analyser les voies métaboliques présentes dans l'échantillon, quelle(s) base(s) de données faut-il utiliser ?

Question 26 Quelles informations au niveau fonctionnelle peut-on aussi tirer des séquences ? Comment faut-il faire ?

4 Comparaison des affiliations taxonomiques des métagénomes entre l'individu sain et l'individu en rémission

La métagénomique comparative consiste à comparer des affiliations taxonomiques et/ou fonctionnels entre plusieurs échantillons afin de comprendre les similarités et les différences et les relier à ce qui différencie les échantillons. Dans ce TP, il faudrait comparer les métagénomes d'un individu sain et d'un individu en rémission et `MEGAN` permet facilement effectuer ces comparaisons.

Tâche Effectuer une analyse de métagénomique comparatives des affiliations taxonomiques des métagénomes de l'individu sain et de l'individu en rémission :

— Importer dans `MEGAN` les sorties `Blast` et les ORFs conservés pour les deux métagénomes en ouvrant à chaque fois une nouvelle session
— Utiliser la commande `Compare` pour comparer les affiliations taxonomiques des deux métagénomes
— Explorer les similarités et les différences à différents niveaux taxonomiques
— Générer des graphiques de comparaison aux différents niveaux taxonomiques

Question 27 A quel niveau taxonomique les différences sont-elles les plus marquées ?

Question 28 Quel est la différence de diversité ? Calculer l'indicateur de diversité β

Question 29 Que dire des différences observées (espèces bactériennes, implication de la présence ou l'absence de certaines bactéries dans la maladie, ...) ?

5 Conclusion

Question 30 Faire un résumé critique du protocole utilisé pour analyser les métagénomes

Question 31 Faire un résumé des résultats obtenus dans l'analyse et la comparaison des affiliations taxonomiques des métagénomes d'un individu sain et d'un individu en rémission d'infections chroniques de l'intestin

Références

[Qin *et al.*, 2010] Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., Mende, D. R., Li, J., Xu, J., Li, S., Li, D., Cao, J., Wang, B., Liang, H., Zheng, H., Xie, Y., Tap, J., Lepage, P., Bertalan, M., Batto, J.-M., Hansen, T., Le Paslier, D., Linneberg, A., Nielsen, H. B., Pelletier, E., Renault, P., Sicheritz-Ponten, T., Turner, K., Zhu, H., Yu, C., Li, S., Jian, M., Zhou, Y., Li, Y., Zhang, X., Li, S., Qin, N., Yang, H., Wang, J., Brunak, S., Doré, J., Guarner, F., Kristiansen, K., Pedersen, O., Parkhill, J., Weissenbach, J., MetaHIT Consortium, Bork, P., Ehrlich, S. D. et Wang, J. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65.