#=================== **Analyse métagénomique avec Kraken2** ===================

# Site de formation officiel : https://ugene.net/metagenomic-classification-with-kraken/

# Base de données Kraken2 : https://benlangmead.github.io/aws-indexes/k2

# Utilisation de Trimmomatic : http://www.usadellab.org/cms/?page=trimmomatic

# Étape 1 : Configuration de l'environnement

# Un environnement est comme une boîte où l'on met tous les outils nécessaires.

# Ici, on ajoute deux sources où chercher les outils : Bioconda et Conda-Forge.

conda config --add channels bioconda

conda config --add channels conda-forge

# On crée une boîte spéciale appelée "metagenomics" où l'on mettra Kraken2 et Trimmomatic.

conda create -n metagenomics kraken2 trimmomatic

# On ouvre cette boîte pour utiliser les outils qu'elle contient.

conda activate metagenomics

# On crée un dossier pour ranger tous nos fichiers de travail.

mkdir metagenomics

cd metagenomics

#============================ Télécharger et préparer Kraken2 ============================

# Kraken2 est un programme qui classe les fragments d'ADN en fonction de leur origine.

# Il a besoin d'une base de données pour comparer les fragments.

# On télécharge une base de données déjà prête, qui fait environ 5 Go.

wget https://genome-idx.s3.amazonaws.com/kraken/minikraken2\_v2\_8GB\_201904.tgz

# On extrait la base de données pour pouvoir l'utiliser.

tar xfvz minikraken2\_v2\_8GB\_201904.tgz

# On supprime le fichier compressé pour libérer de l'espace sur l'ordinateur.

rm minikraken2\_v2\_8GB\_201904.tgz

# On crée un dossier pour ranger les fichiers de données brutes.

mkdir data

ls data # Vérifier que le dossier a bien été créé

#============================ Télécharger les séquences d'adaptateurs pour Trimmomatic ============================

# Trimmomatic est un outil qui nettoie les fichiers d'ADN en supprimant les erreurs.

# Il a besoin de fichiers appelés "adaptateurs" pour bien faire son travail.

wget https://raw.githubusercontent.com/usadellab/Trimmomatic/refs/heads/main/adapters/TruSeq3-PE.fa

#============================ Télécharger les données FASTQ brutes ============================

# Les fichiers FASTQ contiennent les séquences d'ADN issues du séquençage.

# Ici, il faut télécharger ces fichiers à la main depuis Google Drive et les placer dans notre dossier.

# Une fois les fichiers téléchargés, on les déplace dans le dossier "data".

mv \*.fastq.gz data/

ls data # Vérifier que les fichiers sont bien là

#============================ Définir des raccourcis pour les fichiers de lecture ============================

# Plutôt que d'écrire à chaque fois les longs noms des fichiers, on définit des raccourcis.

read1=data/lib3.R1\_001.fastq.gz

read2=data/lib3.R2\_001.fastq.gz

#============================ Exécuter Trimmomatic pour nettoyer les lectures ============================

# On utilise Trimmomatic pour enlever les parties inutiles des séquences et améliorer leur qualité.

trimmomatic PE $read1 $read2 \

lib3.trimmed.paired.R1.fastq.gz lib3.trimmed.unpaired.R1.fastq.gz \

lib3.trimmed.paired.R2.fastq.gz lib3.trimmed.unpaired.R2.fastq.gz \

ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 MINLEN:36

# On range les fichiers nettoyés dans un dossier spécial.

mkdir trimmed

mv \*.gz trimmed/

ls trimmed # Vérifier que tout est bien rangé

#============================ Mettre à jour les raccourcis pour les fichiers nettoyés ============================

# Maintenant, on utilise les fichiers nettoyés au lieu des fichiers bruts.

read1=trimmed/lib3.trimmed.paired.R1.fastq.gz

read2=trimmed/lib3.trimmed.paired.R2.fastq.gz

krakendb=minikraken2\_v2\_8GB\_201904\_UPDATE

#============================ Exécuter Kraken2 pour la classification taxonomique ============================

# On crée un dossier pour stocker les résultats de Kraken2.

mkdir kraken\_report

# On lance Kraken2 pour analyser les fichiers d'ADN et voir à quelles espèces ils appartiennent.

kraken2 --use-names \

--db $krakendb \

--threads 2 \

--report kraken\_report/lib3.report \

--paired --gzip-compressed $read1 $read2 > kraken\_report/lib3.kraken

#============================ Vérifier les sorties ============================

# On regarde si Kraken2 a bien fait son travail en listant les fichiers produits.

ls kraken\_report

# On affiche les premières lignes du rapport pour voir un aperçu des résultats.

head kraken\_report/lib3.report

head kraken\_report/lib3.kraken

#============================ Installer et configurer Krona pour la visualisation ============================

# Krona est un outil qui permet de faire un joli graphique des résultats.

# On commence par le télécharger et l'installer.

cd ~

git clone https://github.com/marbl/Krona.git

cd Krona/KronaTools

sudo ./install.pl

# On ajoute Krona au chemin d'accès pour pouvoir l'utiliser partout.

export PATH=$PATH:~/Krona/KronaTools/bin

# On recharge la session pour que les changements soient pris en compte.

source ~/.bashrc

#============================ Générer une visualisation Krona ============================

# On extrait les informations importantes du rapport Kraken2 pour les mettre dans Krona.

cd ~/metagenomics/kraken\_report

tail -n +2 lib3.report | cut -f2,3,6 > lib3\_krona\_input.txt

# On utilise Krona pour créer un fichier HTML interactif.

ktImportText lib3\_krona\_input.txt -o lib3\_krona.html

# Pour voir le résultat, on ouvre le fichier dans un navigateur web.

explorer.exe lib3\_krona.html # Pour Windows

xdg-open lib3\_krona.html # Pour Linux