

# Outils pour l'Analyse de Séquençage

March 12, 2024

## 1 FastQC Contrôle qualité

### 1.1 Définition

FastQC est un outil utilisé pour évaluer la qualité des données de séquençage.

### 1.2 Explication d'utilisation

Vous pouvez exécuter FastQC sur vos fichiers de données de séquençage pour obtenir un rapport détaillé sur la qualité des données.

### 1.3 Exemple d'utilisation sur Python

```
import subprocess

fastq_file = "chemin/vers/votre/fichier.fastq"
fastqc_command = ["fastqc", fastq_file]
process = subprocess.Popen(fastqc_command, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
stdout, stderr = process.communicate()

if process.returncode == 0:
    print("FastQC termin  avec succ s!")
else:
    print("Erreur lors de l'ex cution de FastQC:", stderr.decode())
    https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
```

## 2 Trimmomatic Nettoyage des reads

### 2.1 Définition

Trimmomatic est un outil utilisé pour nettoyer les reads de séquençage.

## 2.2 Explication d'utilisation

Vous pouvez utiliser Trimmomatic pour éliminer les bases de faible qualité et les adaptateurs des données de séquençage.

## 2.3 Exemple d'utilisation sur Python

```
import subprocess

input_file = "chemin/vers/votre/fichier.fastq"
output_file = "chemin/vers/votre/fichier_nettoye.fastq"
trimmomatic_command = ["trimmomatic", "SE", "-phred33", input_file,
output_file,
"ILLUMINACLIP:adapters.fa:2:30:10", "LEADING:3", "TRAILING:3",
"SLIDINGWINDOW:4:15", "MINLEN:36"]

process = subprocess.Popen(trimmomatic_command, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
stdout, stderr = process.communicate()

if process.returncode == 0:
    print("Trimmomatic termin  avec succ s!")
else:
    print("Erreur lors de l'ex cution de Trimmomatic:", stderr.decode())
    print(http://www.usadellab.org/cms/index.php?page=trimmomatic).
```

# 3 Bowtie2 Alignement des reads sur les g nomes

## 3.1 D finition

Bowtie2 est un outil utilis  pour aligner les reads sur un g nome de r f rence.

## 3.2 Explication d'utilisation

Vous pouvez utiliser Bowtie2 pour mapper les reads de s quen age sur un g nome de r f rence.

## 3.3 Exemple d'utilisation sur Python

```
import subprocess

input_file = "chemin/vers/votre/fichier_reads.fastq"
genome_index = "chemin/vers/votre/genome_index"
output_file = "chemin/vers/votre/fichier_alignements.sam"
bowtie2_command = ["bowtie2", "-x", genome_index, "-U", input_file, "-S",
```

```

output_file]

process = subprocess.Popen(bowtie2_command, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
stdout, stderr = process.communicate()

if process.returncode == 0:
    print("Bowtie2 termin  avec succ s!")
else:
    print("Erreur lors de l'ex cution de Bowtie2:", stderr.decode())
    http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.

```

## 4 Matplotlib Production de visualisation cod  e

### 4.1 D  finition

Matplotlib est une biblioth  que Python utilis  e pour produire des visualisations de donn  es.

### 4.2 Explication d'utilisation

Vous pouvez utiliser Matplotlib pour cr  er des graphiques personnalis  s    partir de vos donn  es de s  quen  age.

```

import matplotlib.pyplot as plt

https://matplotlib.org/.

```