Assemblage de génomes bactériens basé sur des références en ligne de commande

Dans ce document, nous avons fourni des instructions pour réaliser un assemblage basé sur des références de *Vibrio cholerae*. Vous pouvez effectuer les étapes sur les données de séquençage brutes que vous avez générées et copiées sur votre ordinateur (voir les instructions pour *configurer votre répertoire de projets* ci-dessous).

Notes importantes pour suivre ce tutoriel :

- Le texte sur fond gris en `police monospace` représente les commandes à saisir. En général, les commandes ne font qu'une ligne, mais dans ce document, les commandes peuvent visuellement s'étendre jusqu'à la ligne suivante. Nous ajouterons une ligne vide entre les commandes pour indiquer la présence de plusieurs commandes.
- Le texte en gras entouré de **< >** est quelque chose que vous devrez remplacer par votre propre nom de dossier, de chemin d'accès ou d'échantillon.
- Ce tutoriel suppose que vous avez configuré le répertoire de pipeline bioinformatique sur votre ordinateur. Avant de commencer le tutoriel ci-dessous, vérifiez si votre ordinateur possède un dossier appelé `bacpage/` dans le répertoire personnel. Si vous n'avez **pas** ce dossier sur votre ordinateur, suivez les instructions de configuration du pipeline bioinformatique pour installer les fichiers et les logiciels nécessaires avant de continuer.

Ce tutoriel vous guidera dans l'assemblage basé sur des références à partir de données de séquençage d'amplicons. Le processus comprend cinq étapes principales, dont une étape d'évaluation des séquences :

1. Aligner des fichiers FASTQ en paires sur un génome de référence, en générant des fichiers BAM.
2. Appeler les variants dans le fichier BAM par rapport à la référence.
3. Générer une séquence consensus à partir des variants.
4. Masquer les régions recombinantes connues à partir de la séquence consensus.
5. Évaluer la qualité des reads de séquençage bruts et de l'alignement.

Ces étapes sont réalisées à l'aide de la plateforme `snakemake`. Snakemake est un outil permettant de créer des pipelines bioinformatiques reproductibles et modulaires. Chaque tâche d'une analyse, y compris celles mentionnées ci-dessus, peut être écrite comme une étape individuelle d'un pipeline. Snakemake facilite la réalisation des analyses en réduisant le nombre de commandes à saisir, parallélise les étapes sur l'ensemble de vos échantillons et confirme que les étapes ont été réalisées avec succès.

Vous trouverez ci-dessous des instructions pour utiliser snakemake afin d'exécuter l'ensemble du pipeline en une seule fois, ainsi que des instructions (dans l'*annexe*) qui vous guideront à travers chacune des étapes ci-dessus, une par une.

ÉTAPE 0 : Configuration du répertoire de projets [*toujours obligatoire*]

Avant de commencer une analyse bioinformatique, il est conseillé de créer un répertoire spécifique pour l'ensemble de données que vous allez analyser. Ce répertoire est l'endroit où vous exécuterez les analyses et enregistrerez les résultats ; il est appelé *répertoire de projets*. Vous souhaiterez créer votre répertoire de projets dans le dossier `bacpage/` de votre répertoire HOME afin de vous assurer que toutes les données de séquençage et tous les résultats se trouvent au même endroit et que vous pouvez utiliser les mêmes outils de pipeline et les mêmes logiciels.

Nous appellerons le chemin d'accès au dossier `bacpage/` **<sequencing-path>**. Sur la plupart des machines, le **<sequencing-path>** sera `~/bacpage`.

1. Naviguez jusqu'au dossier de pipeline bioinformatique comme décrit ci-dessus :

```
cd ~/bacpage
```

2. Le dossier `bacpage/` contient un sous-dossier appelé `example/`. La première étape consiste à faire une copie de ce dossier. Vous ferez une nouvelle copie chaque fois que vous effectuerez un nouveau séquençage, ce qui vous permettra de vous assurer que les fichiers et les dossiers du répertoire example sont configurés exactement de la même manière à chaque fois. Pour copier ce dossier et lui donner un nom spécifique au projet, exécutez la commande ci-dessous. Nous vous recommandons de donner à votre répertoire de projets un nom informatif, tel que <date>_<sequencing-run-name>. (par exemple : `20220609_cholera_run1`).

   Tapez ce qui suit dans votre fenêtre de terminal et appuyez sur **Entrée** :

```
cp -R example/ <project-directory-name>
```

3. Naviguez jusqu'au répertoire de votre projet et enregistrez le chemin absolu en utilisant.

```
cd <project-directory-name>

pwd
```

   Nous ferons référence au chemin absolu de votre répertoire de projet en tant que **<project-path>** Vous aurez besoin du chemin absolu de votre répertoire de projet à l'*étape 6* ci-dessous.

4. Localisez les données d'entrée que vous souhaitez utiliser dans le pipeline d'assemblage. Ce pipeline nécessite des fichiers FASTQ démultiplexés (c'est-à-dire deux fichiers FASTQ par échantillon). En utilisant l'outil de recherche de fichiers de votre ordinateur, déplacez ces données dans le répertoire `input/` du répertoire de votre projet (c'est-à-dire le dossier que vous venez de créer ci-dessus).

5. Dans votre répertoire de projets, il devrait y avoir un fichier appelé `sample_data.csv`, qui contiendra les informations sur vos échantillons. Ouvrez ce fichier dans Excel ou un tableur similaire et ajoutez les informations relatives à chacun de vos échantillons sur une ligne individuelle.

   Dans la colonne **échantillon**, enregistrez le nom ou l'identifiant de chaque échantillon (ceux-ci doivent être uniques et ne pas contenir de caractères inhabituels tels que des points « . » ou des barres obliques « / »). Dans la colonne **read1**, enregistrez le chemin absolu du fichier FASTQ correspondant au premier ensemble

de reads d'un échantillon (contenant généralement « R1 » dans le nom de fichier). Dans la colonne **read2**, enregistrez le chemin absolu du fichier FASTQ correspondant à la première série de reads pour un échantillon (contenant généralement "R2" dans le nom de fichier).

Le chemin absolu d'un fichier peut être déterminé à l'aide de la commande `pwd`. Si vos fichiers FASTQ démultiplexés ont été placés dans le répertoire `input/` de votre répertoire de projets comme décrit ci-dessus, vous pouvez trouver leur chemin absolu en exécutant la commande suivante :

```
cd <project-path>/input

pwd
```

Le chemin absolu de vos fichiers sera le résultat de `pwd` plus « / » plus le nom du fichier. Un exemple de fichier `sample_data.csv` complété est présenté ci-dessous :

sample	read1	read2
échantillon1	/home/user/bacpage/20220609_run1/input/sample1_S13_L001_R1_001.fastq.gz	/home/user/bacpage/20220609_run1/input/sample1_S13_L001_R2_001.fastq.gz
échantillon2	/home/user/bacpage/20220609_run1/input/sample2_S3_L001_R1_001.fastq.gz	/home/user/bacpage/20220609_run1/input/sample2_S3_L001_R2_001.fastq.gz
échantillon3	/home/user/bacpage/20220609_run1/input/sample3_S22_L001_R1_001.fastq.gz	/home/user/bacpage/20220609_run1/input/sample3_S22_L001_R2_001.fastq.gz

6. Enregistrez le fichier après avoir saisi les données de vos échantillons.

7. Dans votre répertoire de projets, ouvrez le fichier de configuration appelé `config.yaml` à l'aide d'un éditeur de texte de votre choix. Ce fichier contient les paramètres et les options de l'analyse. Les paramètres peuvent être modifiés en fonction de votre analyse, mais les options par défaut devraient convenir à la plupart des analyses.

   Le fichier de configuration étant ouvert dans un éditeur de texte, remplacez **<project-path>** et **<sequencing-path>** par leurs *chemins absolus* pour les six premiers paramètres.

```
# General parameters; point to input files.
run_type: "Illumina"
samples: "<project-path>/sample_data.csv"
output_directory: "<project-path>"
reference: "<sequencing-path>/resources/vc_reference.fasta"
reference_genes: "<sequencing-path>/resources/cholera_ref_genes/"
recombinant_mask: "<sequencing-path>/resources/cholera_mask.gff"
```

8. Enfin, déterminez le nombre de processeurs disponibles sur votre ordinateur, car l'utilisation d'un plus grand nombre de processeurs accélérera l'exécution du pipeline. La sortie de la commande suivante indiquera le nombre de processeurs dont vous disposez. Exécutez la commande ci-dessous et notez le résultat pour plus tard :

```
cat /proc/cpuinfo | grep processor
```

ÉTAPE 1 : Exécuter le pipeline complet

Si vous souhaitez exécuter l'ensemble du pipeline sans passer par chacune des étapes intermédiaires, vous pouvez générer des séquences consensus et calculer des mesures de la qualité à l'aide d'une seule commande.

1. Naviguez jusqu'à l'emplacement du pipeline bioinformatique :

```
cd ~/bacpage
```

2. Exécutez le pipeline à l'aide de la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until mask_consensus
generate_complete_report
```

Cela générera une séquence consensus au format FASTA pour chacun de vos échantillons et les placera dans **<project-path>**/`results/consensus_sequences`/**<sample>**`.masked.fasta`. Un rapport HTML contenant les mesures d'alignement et de qualité pour vos échantillons se trouve dans **<project-path>**/`results/reports/qc_report.html`.

Si la commande de snakemake a été exécutée avec succès, vous devriez voir quelque chose comme ceci à l'écran :

```
[Tue Sep  5 12:09:40 2023]
Finished job 119.
253 of 253 steps (100%) done
Complete log: .snakemake/log/2023-09-05T120148.784555.snakemake.log
```

Note : Le numéro de la tâche, l'étape et le nombre total d'étapes dépendent du nombre d'échantillons que vous avez.

Si une commande de snakemake n'a pas abouti, vous verrez ce qui suit à la fin de la sortie :

```
Error in rule x:
    jobid: 1
    shell:
        command -options arguments
Shutting down, this might take some time.
Exiting because a job execution failed. Look above for error message
Complete log: .snakemake/log/2023-09-05T120148.784555.snakemake.log
```

Enregistrez la règle manquée, l'échantillon en cours de traitement et le fichier de sortie attendu de la règle manquée. Vous pouvez essayer de répéter l'étape sur un échantillon spécifique en exécutant la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going
```

ÉTAPE 2 : Évaluer la qualité de l'assemblage du génome

Un élément clé de la génération d'assemblages de génomes est l'évaluation de leur qualité. Le pipeline bioinformatique génère un rapport permettant d'inspecter visuellement la qualité, la couverture et la confiance que nous avons dans les séquences consensus résultantes.

1. Pour examiner les mesures de qualité de vos échantillons, ouvrez **<project-path>**/`results/reports/qc_report.html` à l'aide d'un navigateur Web.
2. Consultez la fiche de référence de la qualité du génome pour connaître les mesures de qualité attendues.

Annexe à l'assemblage de génomes en ligne de commande

Les instructions suivantes décrivent les étapes intermédiaires de la génération des séquences consensus. Ce processus peut être effectué en option au lieu d'exécuter le pipeline complet à l'aide des instructions ci-dessus.

ÉTAPE 1 : Aligner des fichiers FASTQ en paires sur un génome de référence

La première étape de l'assemblage du génome consiste à prendre les fichiers FASTQ en paires produits par une machine de séquençage et à les aligner sur un génome de référence. Cela permet de générer un fichier BAM pour chaque échantillon, qui est nécessaire pour les étapes suivantes du pipeline.

Note : par défaut, le pipeline alignera les reads sur une référence *Vibrio cholerae*. Si vous étudiez un autre agent pathogène, vous pouvez modifier les paramètres *« de référence »* dans <**`project-path`**>/config.yaml pour une autre séquence de référence.

1. Naviguez dans votre répertoire de travail actuel jusqu'à l'emplacement du pipeline bioinformatique (généralement `~/bacpage`). Toutes les étapes suivantes doivent être effectuées dans ce répertoire. Tapez ce qui suit dans votre fenêtre de terminal et appuyez sur **Entrée** :

```
cd ~/bacpage
```

2. Exécuter l'étape alignment_bwa du pipeline à l'aide de la commande suivante.

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until alignment_bwa
```

   Cette commande aligne individuellement chaque échantillon indiqué dans le fichier `sample_data.csv`, et prendra quelques minutes en fonction du nombre d'échantillons.

   Un fichier BAM sera généré pour chaque échantillon au format <**`sample`**>`.sorted.bam`. Les fichiers BAM pour chaque échantillon se trouvent dans <**`project-path`**>`/intermediates/illumina/alignments/`.

ÉTAPE 2 : Appel des variants par rapport au génome de référence

L'étape suivante de l'assemblage du génome consiste à déterminer les différences entre les reads séquencés et le génome de référence. Ces différences sont appelées variants et décrivent l'évolution de l'échantillon par rapport au génome de référence (généralement le génome le plus ancien d'un organisme).

1. Exécuter l'étape d'appel de variants du pipeline à l'aide de la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until
align_and_normalize_variants
```

Cette commande appelle les variants pour chaque échantillon indiqué dans le fichier `sample_data.csv`, filtre les variants de faible qualité et non pris en charge, et normalise les insertions et les suppressions. Un fichier VCF (Variant Call Format) sera généré pour chaque échantillon au format **<sample>**`.filt.norm.vcf.gz` (voir https://en.wikipedia.org/wiki/Variant_Call_Format pour une description du format VCF). Les fichiers VCF pour chaque échantillon se trouvent dans **<project-path>**`/intermediates/illumina/variants/`

ÉTAPE 3 : Générer une séquence consensus

Nous allons maintenant générer une séquence génomique pour chaque échantillon en appliquant les variants à notre séquence de référence. Comme ce génome est un résumé de plusieurs reads de séquençage, nous appelons cette séquence une séquence consensus.

1. Exécutez l'étape d'appel de consensus du pipeline à l'aide de la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until call_consensus
```

Cette commande génère une séquence consensus pour chaque échantillon indiqué dans le fichier `sample_data.csv`. Un fichier FASTA contenant la séquence consensus sera généré pour chaque échantillon au format **<sample>**`.consensus.fasta`. Les fichiers FASTA pour chaque échantillon se trouvent dans **<project-path>**/intermediates/illumina/consensus/.

ÉTAPE 4 : Masquer la séquence consensus

Nous recommandons de masquer les régions du génome qui ont une faible couverture et/ou qui sont entièrement recombinantes. Le masquage de ces régions permet d'éviter les résultats erronés des analyses en aval, y compris le typage et l'inférence phylogénétique. Les régions à faible couverture peuvent être déterminées directement à partir du fichier BAM généré pour chaque échantillon, tandis que les régions entièrement recombinantes nécessitent une connaissance préalable de l'organisme étudié. Dans le cadre du pipeline, nous avons fourni un fichier indiquant les régions entièrement recombinantes du génome du choléra. Si vous étudiez un autre organisme, vous devrez mettre à jour le paramètre recombinant_mask dans le fichier de configuration avec un autre masque spécifique à l'organisme.

1. Pour exécuter l'étape de masquage du pipeline, utilisez la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until mask_consensus
```

Cette commande masque la séquence consensus pour chacun de vos échantillons. Un fichier FASTA contenant la séquence consensus masquée sera généré pour chaque échantillon au format **<sample>**.masked.fasta. Les fichiers FASTA masqués pour chaque échantillon se trouvent dans : **<project-path>**/intermediates/illumina/consensus/

ÉTAPE 5 : Évaluer la qualité de l'assemblage

Un élément clé de la génération d'assemblages de génomes est l'évaluation de leur qualité. Nous générerons des rapports pour inspecter visuellement la qualité, la couverture et la confiance que nous avons dans les séquences consensus résultantes.

1. Générez les rapports de contrôle de qualité à l'aide de la commande suivante :

```
snakemake --configfile <project-path>/config.yaml --cores
<number-of-processors> --keep-going --until generate_complete_report
```

Cette commande génère un rapport HTML unique contenant les mesures de qualité pour tous les échantillons. Le rapport HTML se trouve dans **<project-path>**/results/reports/qc_report.html.

2. Ouvrez ce fichier avec un navigateur Web.
3. Consultez la fiche de référence pour la qualité du génome pour connaître les mesures de qualité attendues.