README : Pipeline SNAKEMAKE – BLOC 2 - Analyse des données ATACseq 2020-2021

Laëtitia Racine – Juin 2021

L'analyse des données ATACseq a été séparée en 3 blocs d'analyses. Le premier bloc correspond au pipeline que Sophie effectuait en sortie de séquençage pour obtenir des .bam propres. Le deuxième bloc prend en entrée les fichiers .bam obtenus en sortie du bloc 1 et permet de créer les granges annotés ainsi que des graphiques de qualité (qc, nbreads, nbpeaks...). Le troisième bloc prend en entrée les fichiers grange_ann pour créer des graphiques d'analyses (volcano plot, homer...).

~~~~~~

BUT DU BLOC 2

- Analyser la qualité des fichiers .bam.
- Effectuer un downsampling sur les échantillons suivi d'un peak calling.
- Faire l'indexing des .bam pour l'observation via IGV.
- Créer des fichiers granges et les annoter.
- Créer des reports qualité et les graphiques associés.

Selon le Snakefile choisi, on peut procéder l'analyse de bout en bout sur **donneurs séparés** (Snakefile_separated_bloc2) ou en **fusionnant les donneurs** sur les .bam (Snakefile_merged_bloc2). Il est également possible de **tester différents seuils** sur le nombre de reads/peak accepté.

Tous les outputs en reports et graphiques possibles sont indiqués dans le rule_all des Snakefile.

~~~~~~

EQUIPEMENT NECESSAIRE

- environnement linux
- miniconda

L'environnement de travail contenant tous les packages nécessaires au déroulement du pipeline se créé automatiquement à partir d'un fichier yaml fourni en entrée avec le snakefile.

~~~~~~

ARBORESCENCE DES FICHIERS D'ENTREE

L'environnement de travail est stocké dans un dépôt git. Il peut être chargé directement à partir du dépôt ou recréé en suivant l'arborescence ci-dessous:

- A_Initial_data
    - Lien vers un dossier externe au git/dossier de travail comprenant les données brutes (.bam dans un sous-dossier bam_files et Annotation_TSS_pm1kb_int_ex_53utr_ctcf_cpg_histo_gr.rda.). Pour créer le lien, il suffit d'ouvrir un terminal et de taper : ln -s CheminCible NomLien.
- B_Environments
    - ATACMetabo_main_env.yaml
    - ATACMetabo_main_env.locked.yaml
- C_Scripts
    - annotate_grange.R
    - broadPeak_to_csv.R
    - Granges.R
    - peaks_featureCounts.R
    - peaks_filter.R
    - peaks_report.R
    - report_plots_merged.R
    - report_plots_separated.R
- E_Documentation
    - Contient des documents d'explications du pipeline.

Snakefile_merged_bloc2
Snakefile_separated_bloc2
analysis_choice.csv

L'arborescence finale contiendra en plus un dossier D_Analysis. Ce dossier est en effet créé automatiquement lors du lancement du snakefile pour y stocker les résultats.
- D_Analysis
~~~~~~

- bam : contient les liens vers le fichiers bruts avec des noms courts contenant timing, donneur et condition, les .qc des bams et les .bai et .nbreads des fichiers avant downsampling.
- downsampled_bam : contient les downsampled.bam, les .bai et les .nbreads après downsampling.
- genomic_ranges : contient les genomic ranges (gr.rds) annotés et non annotés, et les tableaux .csv annotés et non annotés
- macs2_output : contient tous les fichiers créés par l'outil macs2 autour de broadPeak, les fichiers équivalent au broakPeak mais au format csv ainsi que les fichiers readcount.
- reports : contient tous les fichiers report.csv ainsi que les graphiques .png et/ou .pdf associés.
- touch : fichiers nécessaires au déroulement correct du pipeline. A effacer quand plus besoin de faire tourner le pipeline.

~~~~~~

FICHIERS DE SORTIE POSSIBLES

Lorsqu'on lance le snakefile, on peut choisir les outputs souhaités (reports/graphiques) via le rule_all.

Les reports pouvant être créés par le snakefile sont les suivants :
- **qc_report.csv** : contient les résultats de l'analyse samtools flagstat portant sur la qualité de séquençage des fichiers .bam. QCP pour QC passed et QCF pour QC failed. Normalement, les QC failed ont été enlevés de l'analyse dans le bloc 1.
- **nbreads_report.csv** : contient le nombre de reads total par échantillon avant downsampling et après downsampling.
- **nbpeaks_per_chromosome_report.csv** :contient pour chaque échantillon la liste des chromosomes représentés et le nombre de peaks correspondant, en fonction du threshold appliqués. Il est assez rare, mais tout de même possible, que des chromosomes disparaissent de la liste après application de threshold car il n'y a plus de peaks acceptés à l'intérieur.
- **nbpeaks_nbreads_long_report.csv** et **nbpeaks_nbreads_wide_report.csv** : contiennent la même information mais présentée différemment. Le long_report est au bon format pour être utilisé sous R, c'est ce report qui permet de tracer des graphiques. Le wide_report est présenté pour une lecture directe du tableau, sans nécessité absolue de passer par les graphiques. Pour chaque échantillon et pour chaque threshold, on a la liste de peaks détectés ainsi que le chromosome auquel ils correspondent et leur nombre de read associé.
- **nbpeaks_report.csv** :contient pour chaque échantillon, le nombre de peaks, le nombre moyen de reads par peak ainsi que le pourcentage de peaks perdus en fonction du threshold appliqué sur le nombre de reads/peak.

Pour les graphiques, leur nom renseigne sur ce qu'ils contiennent :
nom du report_format du graphique -information représentée
Attention toutefois, tous les formats de graphique ne sont pas utilisables pour tous les reports. Il faut se référer à la liste disponible dans le rule_all pour savoir quel graphique il est possible de tracer.
Voici une liste des formats proposés :
- hist_donor (donneurs séparés) : histogramme en facet sur les donneurs (un graphique par donneur)
- hist_time (donneurs séparés) : histogramme en facet sur les temps (un graphique par temps)
- hist (donneurs fusionnés) : histogramme sans facet

- line_cond (donneurs séparés) : nuage de point et ligne en facet sur les donneurs
- line_time (donneurs séparés) : nuage de point et ligne en facet sur les temps
- line (donneurs fusionnés) : nuage de point et ligne en facet sur les conditions

- chrom_single (deux analyses) : un graphique par chromosome, on a donc 25 graphiques
-chrom_multi (deux analyses) : tous les chromosomes sur même graphique, un graphique par échantillon

-freq (deux analyses) : distribution du nombre de reads par peak

~~~~~~

EXPLICATION DU PIPELINE

- Le fichier analysis_choice.csv permet de choisir les échantillons à analyser. Un exemple est disponible dans le dossier E_Documentation (analysis_choice_example.csv).
- L'explication des rules du snakefile est disponible dans le document Explication_Snakefile. Le documentation a été créé à partir du Snakefile_merged_bloc2 mais s'applique aux règles du Snakefile_separated_bloc2 car les documents sont très proches.
- Les scripts R sont commentés entre les lignes de code lorsque c'est nécessaire.

~~~~~~

LANCEMENT DU PIPELINE
~~~~~~

Préparation de l'espace de travail et choix d'analyses
- Créer un dossier de travail en suivant l'arborescence décrite ou le charger depuis le git.
- Mettre tous les fichiers bruts .bam ainsi que le tableau d'annotations dans un dossier raw data externe au dossier de travail et faire un lien vers ce dossier dans le dossier de travail nommé A_Initial_data
- Vérifier que tous les fichiers d'entrée sont correctement rangés dans l'arborescence.
- Choisir le mode de travail (donneurs séparés ou fusionnés) et ouvrir le Snakefile correspondant.
- Indiquer les thresholds souhaités pour le nombre de reads par peak dans la variable THRESHOLD_TO_TEST en haut du Snakefile (ligne 9) et enregistrer le fichier.
- Commenter/Décommenter (ctrl+alt+/) les outputs souhaités dans la rule_all et enregistrer le fichier.
- Remplir le fichier analysis_choice.csv (ou le créer à partir de l'exemple disponible dans E_Documentation) en indiquant TRUE/FALSE dans la colonne INCLUDED selon l'analyse souhaitée.
!!! ATTENTION, étape importante => la valeur de downsampling et donc les résultats finaux peuvent varier selon les échantillons sélectionnés !!!
- Ouvrir un terminal depuis le dossier de travail.

Lancement du script
Pour tous les exemples suivants, on utilisera Snakefile_separated _bloc2 et cores 12 mais on peut évidemment utiliser plus/moins de coeurs selon l'ordinateur disponible et les lignes de commandes s'appliquent également pour Snakefile_merged_bloc2.

Pour faire tourner le script et copier l'affichage de la console dans un fichier texte :
script Snakefile_separated_bloc2__date.txt
snakemake -s Snakefile_separated_bloc2 --use-conda --cores 12--reason
exit

Pour faire un report du pipeline :
snakemake -s Snakefile_separated_bloc2 --use-conda --cores 12 --reason --report Snakefile_separated_bloc2_ date.html

Pour faire un dag du pipeline :
snakemake -s Snakefile_separated_bloc2 –rulegraph | dot -Tpdf > Snakefile_separated_bloc2_dag.pdf

ATTENTION ! Il est possible de lancer plusieurs fois le même Snakefile dans le même dossier avec des choix différents dans analysis_choice.csv. Les fichiers seront recréés seulement si quelque chose change (si même valeur de downsampling, il n'est pas refait). Toutefois, si on souhaite lancer l'autre Snakefile, il faut travailler dans un nouveau dossier vide.

~~~~~~

TRANSMETTRE LE PIPELINE

Possibilité n°1 :
- Fournir le lien du git pour qu'il puisse récupérer le dernier commit : il aura donc accès à tous les documents des dossiers B_Environments, C_Scripts, E_Documentation ainsi que les Snakefile_merged_bloc2 et Snakefile_separated_bloc2.
- Envoyer le fichier Annotation_TSS_pm1kb_int_ex_53utr_ctcf_cpg_histo_gr.rda.
- Dire de suivre ce document pour remplir son fichier analysis_choice.csv et lancer le pipeline.

Possibilité n°2 :
- Envoyer les documents contenus dans les dossiers B_Environments, C_Scripts, E_Documentation.
- Envoyer le fichier Annotation_TSS_pm1kb_int_ex_53utr_ctcf_cpg_histo_gr.rda.
- Envoyer les fichiers Snakefile_merged_bloc2 et Snakefile_separated_bloc2.
- Dire d'ouvrir ce document d'explication et de créer l'arborescence comme indiquée.
- Dire de créer le fichier analysis_choice à partir de l'exemple disponible dans E_Documentation.
- Dire de lancer le snakefile comme indiqué dans ce document.
~~~~~~