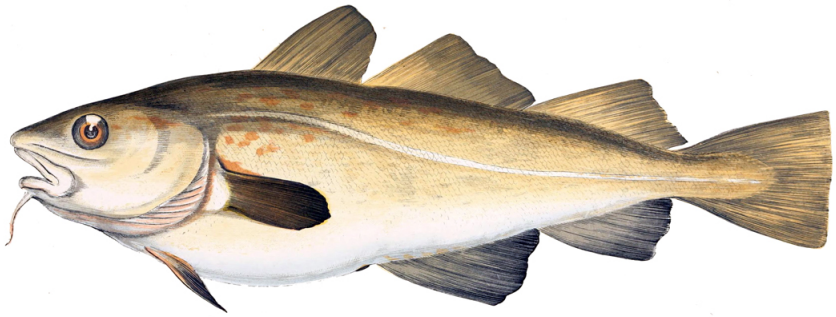
Projet Bioinformatique – Mise en œuvre du pipeline Nextflow nf-core/rnavar sur des données NCBI de Gadus morhua

Camille-Astrid Rodrigues

October 2025

Biologie des Systèmes
Université de Toulouse



1 Introduction

L'analyse des données de séquençage haut-débit est devenue un outil essentiel pour explorer la diversité génétique et fonctionnelle des organismes. Parmi les méthodes les plus couramment utilisées, le séquençage d'ARN (RNA-seq) permet de mesurer l'expression des gènes, de détecter des variants et de mieux comprendre les mécanismes moléculaires sous-jacents. Dans ce cadre, les pipelines de type *workflow* bioinformatique, tels que ceux proposés par la communauté `nf-core`, offrent une solution reproductible, portable et standardisée pour le traitement de ces données.

Le présent projet a pour objectif de mettre en œuvre le pipeline **nf-core/rnavar**, développé pour l'analyse de variants à partir de données RNA-seq. L'exercice se déroule sur le cluster de calcul `genobioinfo`, en utilisant l'environnement Nextflow associé aux modules nécessaires (Java, Singularity/containers, et workflows nf-core).

Nous avons choisi comme organisme d'étude la morue de l'Atlantique (*Gadus morhua*), espèce modèle importante pour les études de biologie marine et de génomique évolutive. Trois échantillons RNA-seq ont été retenus à partir de la base de données NCBI SRA :

- SRR2045415 : échantillon issu de tissu ovarien,
- SRR2045416 : échantillon issu de tissu cérébral,
- SRR2045417 : échantillon supplémentaire du même projet.

Le but est de suivre l'ensemble de la démarche bioinformatique : de la récupération des données brutes et des fichiers d'annotation, à la mise en place de l'environnement d'exécution sur le cluster, jusqu'au lancement du pipeline `nf-core/rnavar`. Les résultats obtenus seront ensuite interprétés afin de documenter les performances et les éventuelles difficultés rencontrées lors de l'exécution.

2 Matériel et Méthodes

2.1 Connexion au cluster de calcul

Toutes les analyses ont été réalisées sur le cluster `genobioinfo`, dans un compte de formation dédié. Le compte utilisé pour l'analyse est le compte fmt101, accessible depuis la connexion :

```
ssh fmt101@genobioinfo.toulouse.inrae.fr
```

Les tâches ont été soumises via le gestionnaire de ressources SLURM (`sbatch`), avec des contraintes de temps et de mémoire adaptées en fonction des étapes du pipeline. Les modules nécessaires ont été chargés à l'aide de la commande `module load`. En particulier :

- `bioinfo/SRA-Toolkit/3.0.2` pour la récupération et la conversion des données SRA en fichiers FASTQ,

- `bioinfo/Nextflow/25.04.0` et `devel/java/17.0.6` pour l'exécution du pipeline `nf-core/rnavar`,
- modules additionnels pour la gestion des containers Singularity et l'environnement Python si nécessaire.

2.2 Jeux de données

Les données utilisées proviennent de la base NCBI Sequence Read Archive (SRA). Nous avons sélectionné trois échantillons RNA-seq de l'espèce *Gadus morhua* (morue de l'Atlantique), correspondant à différents tissus :

- **SRR2045415** : tissu ovarien, 22.4 M spots (4.5 Gb),
- **SRR2045416** : tissu cérébral, 36.5 M spots (7.3 Gb),
- **SRR2045417** : troisième échantillon complémentaire du projet.

2.3 Téléchargement et conversion des données

La récupération des données a été réalisée en deux étapes :

1. Téléchargement des fichiers `.sra` via la commande `prefetch`,
2. Conversion au format FASTQ compressé (`.fastq.gz`) avec `fastq-dump --split-files --gzip`.

Un script automatisé (`load_data.sh`) a été rédigé afin de soumettre ces opérations au cluster :

```
#!/bin/bash
#SBATCH -p workq
#SBATCH --time=1-11:00:00 #1 jour et 11h

#Load binaries
module purge
module load bioinfo/SRA-Toolkit/3.0.2

vdb-config --prefetch-to-cwd;
prefetch SRR2045415;
prefetch SRR2045416;
prefetch SRR2045417;
fastq-dump --gzip --split-files SRR2045415;
fastq-dump --gzip --split-files SRR2045416;
fastq-dump --gzip --split-files SRR2045417;

mkdir fasta;
mkdir genome;
mkdir annotation;
```

```
mkdir output;

cd annotation;

wget https://ftp.ensembl.org/pub/release-115/gtf/gadus_morhua/
Gadus_morhua.gadMor3.0.115.gtf.gz ;

gunzip Gadus_morhua.gadMor3.0.115.gtf.gz ;
rm Gadus_morhua.gadMor3.0.115.gtf.gz ;

cd .. ;
mv SRR* fasta/ ;

mv slurm* output/ ;
```

2.4 Organisation des répertoires

Afin de structurer le projet, différents dossiers ont été créés :

- `fasta/` : contient les fichiers FASTQ compressés,
- `annotation/` : contient les fichiers d'annotation (GTF),
- `genome/` : contient le génome de référence au format FASTA,
- `output/` : contient les résultats et les fichiers de log.

2.5 Fichiers d'annotation

Les fichiers d'annotation génomique ont été téléchargés depuis Ensembl (release 115). Le fichier GTF correspondant au génome `gadMor3.0` a été récupéré puis décompressé via le pipeline `load_data.sh`.

2.6 Préparation du fichier samplesheet

Le pipeline `nf-core/rnavar` nécessite un fichier CSV décrivant les échantillons et les chemins vers les fichiers FASTQ. Un fichier `samplesheet.csv` a donc été généré, avec la structure suivante :

```
sample,fastq_1,fastq_2,strandedness
TESTONE,/path/to/SRR2045415_1.fastq.gz,/path/to/SRR2045415_2.fastq.gz,auto
TESTTWO,/path/to/SRR2045416_1.fastq.gz,/path/to/SRR2045416_2.fastq.gz,auto
TESTTHREE,/path/to/SRR2045417_1.fastq.gz,/path/to/SRR2045417_2.fastq.gz,auto
```

2.7 Exécution du pipeline Nextflow

L'exécution du pipeline `nf-core/rnavar` a été effectuée en soumettant un script SLURM (`pipeline_rnavar.sh`) qui charge les modules nécessaires et lance le workflow :

```
#!/bin/bash
#SBATCH -J nfcorernaseq
#SBATCH -p unlimitq
#SBATCH --mem=6G

module load bioinfo/Nextflow/25.04.0
module load devel/java/17.0.6

input=/home/fmt101/work/Project/samplesheet.csv
gtf=/home/fmt101/work/Project/annotation/Gadus_morhua.gadMor3.0.115.gtf

nextflow run nf-core/rnavar -profile genotoul -r 1.2.1 \
    --input $input \
    --outdir output/OUTDIR_rnavar \
    --gtf $gtf \
    --skip_baserecalibration
```

3 Discussion

Au cours de la réalisation de ce projet, plusieurs difficultés techniques ont été rencontrées lors de la mise en place de l'environnement, du téléchargement des données et du lancement du pipeline. Cette section discute ces problématiques, les solutions mises en place et les perspectives d'amélioration.

3.1 Téléchargement des données SRA

La première difficulté rencontrée concernait l'utilisation de la version du `SRA Toolkit` disponible sur le cluster. L'emploi de la version `bioinfo/sratoolkit.2.8.2-1` a conduit à un échec de job (erreur 127), car le module n'était plus disponible ou obsolète. La solution a consisté à rechercher la version correcte du logiciel avec `search_module`, puis à utiliser le module `bioinfo/SRA-Toolkit/3.0.2`, ce qui a permis de lancer correctement les commandes `prefetch` et `fastq-dump`.

Une autre difficulté concernait l'utilisation de liens symboliques vers les banques NCBI (`bank/ncbi`). Les permissions sur le cluster ne permettaient pas la création de ces liens. Le problème a été contourné en téléchargeant directement les fichiers SRA dans le répertoire de travail, puis en les convertissant en FASTQ compressés (`.fastq.gz`) à l'aide de `fastq-dump`.

Le code ci-dessous retrace les indications issues du site `GenoToul Bioinfo` (https://bioinfo.genotoul.fr/index.php/faq/bioinfo_tips_faq/?highlight=sratoolkit).

```
mkdir ncbi
cd ncbi
ln -s ~/work/Project/ncbi/ ~/../../bank/ncbi/
-------------------------------------------------
ln: failed to create symbolic link '/home/fmt101/../../bank/ncbi/ncbi':
Permission denied
```

```
module load bioinfo/sratoolkit.2.8.2-1
-------------------------------------------------
ERROR: Unable to locate a modulefile for 'bioinfo/sratoolkit.2.8.2-1'
```

Après un échec des premières étapes, nous avons recherché le module actuellement disponible sur la plateforme.

```
search_module sratoolkit
search_module SRA
-------------------------------------------------
bioinfo/SRA-Toolkit/3.0.2
bioinfo/Transrate/1.0.3
bioinfo/Transrate/1.0.3_for_drap
```

On maintenant suivre les indications sur le `dépôt GitHub` (https://github.com/ncbi/sra-tools/wiki/08.-prefetch-and-fasterq-dump#how-to-use-prefetch-and-fasterq-dump-to-extract-fastq-files-from-sra-run-accessions)

```
module load bioinfo/SRA-Toolkit/3.0.2
vdb-config --prefetch-to-cwd
prefetch SRR2045415
-------------------------------------------------
2025-10-03T09:43:21 prefetch.3.0.2: Current preference is set to retrieve
SRA Normalized Format files with full base quality scores.
2025-10-03T09:43:21 prefetch.3.0.2: 1) Downloading 'SRR2045415'...
2025-10-03T09:43:21 prefetch.3.0.2: SRA Normalized Format file is being
retrieved, if this is different from your preference, it may be due to
current file availability.
2025-10-03T09:43:21 prefetch.3.0.2:  Downloading via HTTPS...
2025-10-03T09:45:46 prefetch.3.0.2:  HTTPS download succeed
2025-10-03T09:45:50 prefetch.3.0.2:  'SRR2045415' is valid
2025-10-03T09:45:50 prefetch.3.0.2: 1) 'SRR2045415' was downloaded
successfully
```

3.2 Gestion du temps et des ressources

Lors des premiers tests, certains jobs soumis avec SLURM ont échoué car le temps et la mémoire demandés étaient trop faibles ou mal ajustés. L'utilisation de la commande `seff <job_id>` a permis de diagnostiquer les ressources réellement consommées. Une révision des paramètres SLURM (allongement du temps maximum et allocation mémoire de 2 Go par cœur) a permis de compléter avec succès les étapes de téléchargement et de conversion.

3.3 Fichiers d'annotation

Le pipeline `nf-core/rnavar` nécessite un fichier d'annotation génomique au format GTF. Dans un premier temps, aucun fichier adéquat n'était disponible

directement dans le cluster. La solution a consisté à télécharger le fichier `Gadus_morhua.gadMor3.0.115.gtf.gz` depuis le site FTP d'Ensembl (release 115), puis à le décompresser et à le placer dans un répertoire `annotation/`. Cette étape a été réalisée avec succès, mais souligne la nécessité de toujours vérifier la compatibilité entre le génome de référence et les annotations fournies au pipeline.

Dans la recherche des paramètres sur le site nf-core/rnavar, il y a une option `-genome` dans la commande `Nextflow run rnavar`. Cependant, on ne trouve pas la clef de notre génome d'intérêt. Nous allons donc devoir utiliser le .gtf .

Il faut donc trouver comment récupérer le fichier.

https://www.unthsc.edu/school-of-biomedical-sciences/wp-content/uploads/sites/13/RNA-Seq_Hu.pdf

Le lien ci-dessus propose de récupérer le gtf file depuis Ensembl. Depuis le site Ensembl , on peut récupérer le numéro du génome de Gadus morhua (gadMor3.0). On peut également récupérer le lien permet l'accès au fichier gtf. (https://www.ensembl.org/info/about/species.html)

3.4 Configuration du pipeline nf-core/rnavar

Lors de la préparation du fichier `samplesheet.csv`, la difficulté principale était de respecter le format attendu par `nf-core/rnavar`. Une erreur de syntaxe dans ce fichier aurait empêché le lancement du pipeline.

La documentation de GenoToul a fourni un exemple de script batch fonctionnel, non adapté ici au cas du projet.

Cependant, lors du lancement de la pipeline, nous avons rencontré diverses erreurs :

```
    WARN: The following invalid input values have been detected:

* --max_cpus: 48
* --max_memory: 120 GB
* --max_time: 4d
* --igenomesIgnore: true
```

L'erreur précédente est due à un problème de configuration du mode sbatch dans le script.

```
WARN: Found the following unidentified headers in
/home/fmt101/work/Project/samplesheet.csv:
        - strandedness

Known sites are required for performing base recalibration.
Supply them with either --dbsnp and/or --known_indels or disable
 base recalibration with --skip_baserecalibration
```

Nous avons alors ajouté au script l'option `--skip_baserecalibration`.

```
WARN: Found the following unidentified headers in
/home/fmt101/work/Project/samplesheet.csv:
        - strandedness

Missing ‘fromPath‘ parameter -- Check script
’/home/fmt101/.nextflow/assets/nf-core/rnavar/subworkflows/local/prepare_genome
/main.nf’ at line: 53 or see ’.nextflow.log’ file for more details
[-        ] NFC...REPARE_GENOME:GUNZIP_FASTA -
```

Nous n'avons pas pu identifier dans le temps imparti les deux erreurs précédentes :

- la reconnaissance du champs `strandedness`
- le paramètre manquant `fromPath`

3.5 Perspectives et recommandations

Les principaux problèmes rencontrés (versions de modules, liens symboliques, fichiers d'annotation manquants) ont pu être résolus par :

- la recherche de modules disponibles avec `search_module`,
- le téléchargement manuel de fichiers d'annotation depuis Ensembl,
- l'utilisation de scripts automatisés pour limiter les erreurs de saisie.

Néanmoins, plusieurs améliorations sont possibles :

- automatiser davantage la gestion des dépendances en utilisant un conteneur Singularity/Docker complet fourni par nf-core,
- documenter systématiquement les versions de logiciels et bases utilisées pour assurer la reproductibilité,
- prévoir une allocation de ressources plus large lors des premiers tests pour éviter des échecs liés à SLURM.

4 Conclusion

Ce travail avait pour objectif de déployer le pipeline `nf-core/rnavar` sur des données RNA-seq de *Gadus morhua* en environnement HPC (`genobioinfo`). Nous avons (i) sélectionné et récupéré trois jeux de données SRA (SRR2045415, SRR2045416, SRR2045417), (ii) automatisé leur conversion en FASTQ compressés avec le `SRA Toolkit` (`prefetch` + `fastq-dump`), (iii) organisé l'arborescence de travail, et (iv) préparé les annotations (GTF Ensembl, gadMor3.0). L'exécution a été menée via SLURM et Nextflow, en s'appuyant sur les modules du cluster.

Plusieurs difficultés ont été rencontrées puis résolues : indisponibilité de versions anciennes de logiciels (remplacées par `bioinfo/SRA-Toolkit/3.0.2`), impossibilité de créer des liens symboliques dans les banques partagées (contournée

par un téléchargement local), ajustements des ressources SLURM (`seff`) et prise en compte de l'annotation via Ensembl. D'autres points restent ouverts : l'avertissement sur la colonne `strandedness` (non attendue par `rnavar` dans ce contexte) et l'erreur `fromPath` liée à la préparation du génome. À court terme, deux axes de déblocage sont envisagés : (1) fournir explicitement le FASTA de référence compatible avec le GTF (`--fasta` et indices générés par le pipeline ou `--save_reference`) plutôt que `--genome` non disponible pour *G. morhua*, et (2) conformer strictement le *samplesheet* au schéma `nf-core/rnavar` (supprimer les champs non pris en charge). Le recalibrage des bases pourra être réactivé ultérieurement en fournissant des sites connus (`--dbsnp`/`--known_indels`) ou conservé désactivé via `--skip_baserecalibration` selon les objectifs.

Au-delà de ces ajustements, le projet a permis de mettre en place une chaîne reproductible de récupération et de préparation de données, directement transposable à d'autres espèces. Pour renforcer la portabilité et la robustesse, nous recommandons l'usage systématique des profils `nf-core` avec conteneurs (Singularity/Docker), la documentation précise des versions logicielles et la validation continue du *samplesheet* et des paramètres via la documentation `nf-core`. Une fois le couple FASTA/GTF finalisé et la configuration `rnavar` stabilisée, l'analyse des variantes et des rapports (incluant MultiQC) pourra être conduite et interprétée de manière fiable sur l'ensemble des échantillons étudiés.

5 Accès aux ressources

L'ensemble des documents, scripts et fichiers utilisés pour ce projet est disponible en ligne afin de garantir la reproductibilité de l'analyse.

Dépôt du projet

Un dépôt contenant l'ensemble des scripts et documents de ce rapport est accessible à l'adresse suivante : https://github.com/CamilleAstrid/fr.utoulouse.BioSyst.NextFlow

Ce dépôt inclut également une version PDF de ce rapport, les fichiers de configuration SLURM, ainsi que les résultats partiels de l'exécution du pipeline.

- https://bioinfo.genotoul.fr/
- https://nf-co.re/