Reprohackathon : Analyse de reproductibilité d'Article

Auteurs :
Michel BOISSAC
Habib DIAGNE
Paul ROUSSEAU
Kamenan N'GADI

Professeurs :
Frederic LEMOINE
Thomas COKELAER

5 décembre 2023

Table des matières

Chapitre 1

Introduction

1.1 Reproductibilité

En sciences, la reproductibilité est la capacité à reproduire les résultats d'une expérience, d'une étude, de façon indépendante. Cela signifie que d'autres chercheurs, utilisant des méthodes similaires et travaillant avec les mêmes données et conditions expérimentales, devraient être en mesure d'obtenir des résultats similaires ou identiques. La reproductibilité est fondamentale pour s'assurer de la fiabilité et de la robustesse des résultats, et s'inscrit donc pleinement dans le cadre de la méthode scientifique. Les aspects clés de la reproductibilité incluent, d'une façon générale, la transparence dans la méthodologie, et la documentation détaillée des protocoles expérimentaux. Dans le cas d'analyses informatiques, la reproductibilité englobe notamment la gestion appropriée des dépendances logiciels, et la disponibilité des codes sources.

Bien qu'elle ait émergée au XVIIe siècle, cette notion de reproductibilité est actuellement au centre des préoccupations scientifiques. En effet, depuis une quinzaine d'années, il a été constaté qu'un nombre significatif de résultats d'études ne pouvaient pas être reproduits de manière fiable par d'autres chercheurs. Cette « crise de la reproductibilité » soulève des inquiétudes quant à la validité et la fiabilité des découvertes, et questionne donc l'avancement global de la connaissance scientifique. Pour corriger cette tendance, un travail méthodologique croissant est réalisé par la communauté scientifique.

Dans le domaine biologique en particulier, les modélisation et les analyses impliquent souvent des méthodes informatiques mises en œuvre par des logiciels dépendant fortement des infrastructures sous-jacentes (*i.e.* les environnements logiciels, clusters de calcul, langages et paradigmes de programmation, *etc*). Dans le cadre de notre travail, nous nous intéressons justement à la reproductibilité des analyses informatiques de l'étude biologique « Intracellular Staphyloccus aureus persisters upon antibiotic exposure », publiée dans *Nature* en 2020.

1.2 Contexte biologique

Lorsqu'elles sont exposees a un stress important, e.g. un traitement antibiotique, certaines bacteries peuvent devenir « persistantes » : elles survivent et deviennent multi-tolerantes a d'autres stresses, en entrant dans un etat de dormance (i.e. un etat metabolique ralenti sans division cellulaire). A la difference des bacteries dites « resistantes », la tolerance n'est pas liée à des mutations genetiques (acquises ou héritées), et n'est donc pas transmise à la descendance cellulaire. De plus, cet etat persistant est reversible a l'arret du stress : une fois la pression levée, les bacteries reprennent leur metabolisme normal et leurs divisions cellulaires. Dans le cadre d'un traitement antibiotique, ces bactéries persistantes pourraient ainsi être à l'origine de réservoirs infectieux dormants, réactivés à l'arrêt du traitement. La persistance bactérienne joue donc potentiellement un rôle clé dans la chronicité de certaines infections, contribuant aux échecs thérapeutiques. L'article que nous étudions vise à caractériser les processus à l'origine de cet état de persistance, en comparant l'expression génétique entre persistants et non persistants chez le staphylocoque doré. L'étude est découpée en différentes étapes :

1. Dans un premier temps, les chercheurs ont pu caractériser les bactéries persistantes par différents critères.
2. Ils ont induit des populations de staphylocoques persistants, identifiés comme tels grâce à ces critères.
3. Leur profil transcriptomique a été analysé et comparé à celui de staphylocoques non persistants.
4. Les mécanismes d'activation et désactivation de la persistance ont ensuite été étudiés.

5. Enfin, les deux étapes précédentes ont permis de caractériser en détail les dynamiques d'expression génétique des persistants. De là, les chercheurs ont identifié les adaptations métaboliques qui en découlent, et ont proposé un modèle général de la régulation de la persistance chez S. aureus.

L'étape 3. est une analyse informatique correspondant à une suite de traitements bioinformatiques et statistiques. Les résultats de cette analyse comprennent notamment deux MA-plots montrant les différences d'expression génétique entre les staphylocoques persistants et non persistants : un réalisé sur l'ensemble des gènes de *S. aureus*, et l'autre focalisé sur les gènes impliqués dans la traduction (*cf* figures suivantes).

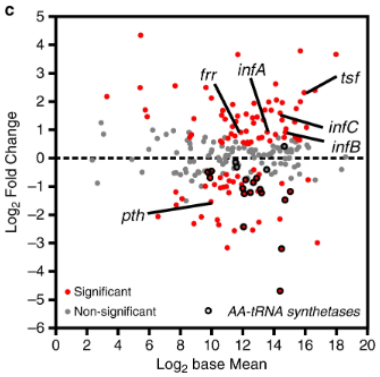

FIGURE 1.1 – MA-plot des gènes impliqués dans la traduction

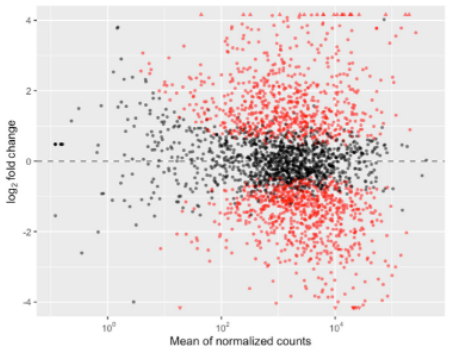

FIGURE 1.2 – MA-plot de tout le jeu de données (tous le

Notre travail a consisté à reproduire cette étape 3., en partant des données de séquençage de l'étape 2. accessibles en ligne. Le but était de suivre le protocole du pipeline d'analyses pour réaliser les deux graphiques ci-dessus.

Chapitre 2

Méthodes et Matériels

Dans le cadre du projet ReproHackathon, nous avons cherché à reproduire les analyses informatiques de l'article à partir des données biologiques en libre accès et du protocole décrit. Pour cela, nous avons implémenté un workflow. Il s'agit du séquençage de l'analyse en différentes étapes bien délimitées et liées entre elles. Ici, chacune des étapes de traitement utilise un environnement logiciel propre, avec ses versions et dépendances particulières. Cela peut être un frein à la reproductibilité des analyses, car il faut s'assurer d'avoir le bon environnement logiciel à chaque étape de la chaîne de traitement, sans quoi les résultats peuvent différer. Pour résoudre ce problème, nous avons donc conteneurisé les étapes du workflow : chaque application utilisée est « encapsulée » dans un fichier contenant tous les éléments nécessaires à son installation, le bon environnement, les dépendances, ..., puis à son exécution.

2.1 Données

Les données utilisées correspondent à six fichiers .fastq, téléchargeables avec sra-toolkit aux adresses SRA suivantes : SRR10379721, SRR10379722, SRR10379723, SRR10379724, SRR10379725, SRR10379726.

Chaque fichier correspond au séquençage du transcriptome d'un échantillon de population bactérienne *S. aureus*. Il y a trois échantillons pour des populations de persistants induits, et trois échantillons de contrôle (non persistants) associés.

Le transcriptome d'une cellule ou d'un ensemble de cellules (ici d'une population bactérienne) correspond à l'ensemble des ARN transcrits à un instant t. Or, plus un gène est transcrit, plus il est exprimé (et inversement). La quantification du transcriptome nous renseigne donc sur le profil d'expression génétique de l'entité séquencée.

2.2 Logiciels utilisés

2.2.1 Nextflow

version 23.10.0

Logiciel d'implémentation de workflow scientifique, particulièrement utilisé pour les analyses bio-informatiques. Il permet d'organiser le processus en segmentant les étapes, tout en les mettant en correspondance. Chaque étape fait généralement appel à un logiciel particulier, et c'est le cas pour notre analyse.

La version de Nextflow utilisée est déjà implémentée dans les VM (machines virtuelles) Biopipes de CloudIFB.

2.2.2 Docker

version 24.0.7

Logiciel permettant d'encapsuler une version d'un ou plusieurs logiciels dans un container, avec les paquets adéquats. Nextflow est capable d'utiliser l'environnement des containers afin de réaliser les scripts des différents processus du workflow.

La version de Docker utilisée est déjà implémentée dans les VM Biopipes de CloudIFB.

2.2.3 R

version 4.3.0

Logiciel de traitement statistique. Dans le cadre de notre étude, nous nous servons du package ***Deseq2 (version 1.4.2)*** pour tester la différence d'expression des gènes entre les échantillons de test et les échantillons de persistants (cf section suivante).

Nous les avons conteneurisés grâce à docker avec les librairies et les dépendances dont nous avions besoin. Le conteneur est disponible sur dockerhub à l'adresse hd88/dsq.

2.2.4 Bowtie

version 0.12.7

Logiciel utilisé pour l'alignement et l'analyse de séquences génétiques. Dans notre étude, nous l'utilisons pour indexer le génome de référence de S. aureus, puis pour l'étape de "mapping" (cf section suivante).

Nous l'avons conteneurisé grâce à Docker avec ses dépendances. Le conteneur est disponible sur DockerHub à l'adresse hd88/bowtie.

2.2.5 Trimgalore

version 0.6.7

Logiciel utilisé pour le nettoyage des données de séquençage génétique. Il permet d'éliminer les séquences ne répondant à certains critères de qualité fixés.

Nous l'avons conteneurisé avec ses dépendances grâce à Docker. Le conteneur est disponible sur DockerHub à l'adresse hd88/trim2, mais il ne fonctionne pas, donc nous avons utilisé un conteneur préexistant créé par le laboratoire VIB (Center for Computational Biology & AI)

2.2.6 Feature Counts

version 2.0.0

Feature Counts est un logiciel qui permet de créer des matrices de comptage des gènes à partir des fichiers .Bam générés à l'étape de mapping.

Nous l'avons conteneurisé avec ses dépendances grâce à docker. Le conteneur est disponible sur DockerHub à l'adresse hd88/fc.

2.3 Description des étapes de l'analyse

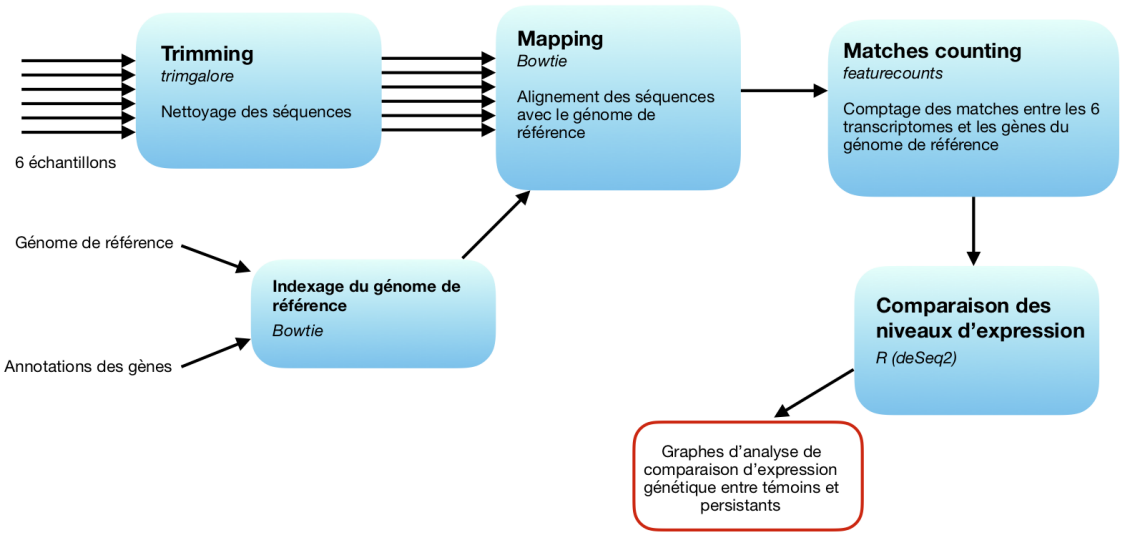

FIGURE 2.1 – Représentation schématique de notre workfow

2.3.1 Téléchargement des données

2.3.2 Trimming

Dans la technique de séquençage, chaque ARN est séquencé plusieurs fois sur des longueurs variables. À chaque ARN sont donc associées plusieurs séquences de longueurs variables. De plus, des erreurs de séquençage sont fréquentes, et un indice de qualité est donc associé à chaque séquence.

Dans l'étape du trimming, pour chaque échantillon, les séquences de moins de 25 nucléotides sont éliminées. En effet, il est statistiquement établi que les motifs sont spécifiques à un emplacement unique du génome au-delà de 25 nucléotides. En-dessous de ce seuil, nous pourrions statistiquement les retrouver à plusieurs positions du génome. On élimine aussi les séquences de « mauvaise qualité » : on se fixe un seuil d'indice de qualité en-dessous duquel les séquences sont considérées comme trop peu fiables par rapport à l'ARN d'origine.

FIGURE 2.2 – Illustration du processus de trimming

2.3.3 Indexage du génome de référence

La séquence brute du génome de référence de S. aureus est téléchargeable à l'adresse suivante : "https ://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi ?db=nuccoreid=CP000253.1rettype=fasta"

On télécharge aussi une base d'annotations correspondant à l'emplacement des gènes, que l'on va utiliser pour indexer ce génome de référence (adresse : "https ://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi ?db=nuccorereport=gff3id=CP0
)

FIGURE 2.3 – Indexage du génome de référence

2.3.4 mapping des séquences avec le génome de référence

Pour chaque échantillon, on va faire coïncider les séquences (après le trimming de la première étape) avec le génome annoté obtenu. Chaque séquence du transcriptome est alignée avec sa séquence correspondante sur l'ADN du génome de référence de *S. aureus*. Au niveau des zones d'introns des gènes, il ne devrait pas y avoir de séquences du transcriptome alignées, mais cette subtilité n'est pas représentée sur le schéma explicatif suivant où nous n'avons pas représenté la présence d'introns.

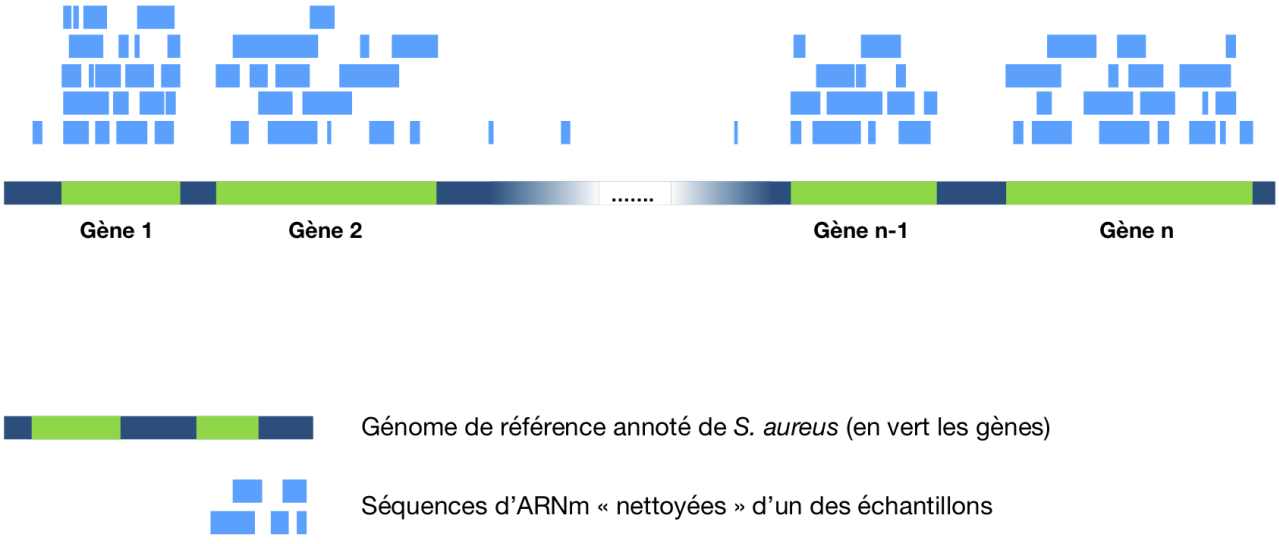

FIGURE 2.4 – Illustration du mapping d'un des échantillons

2.3.5 Comptage des matches

Pour chaque échantillon, on va compter combien de séquences du transcriptome sont associées à chaque gène du génome de référence après leur alignement. Ces valeurs sont consignées dans une matrice, avec une colonne par échantillon et une ligne par gène du génome de référence. Dans la cellule à la ligne i et colonne j, on trouvera donc le nombre de matches entre le séquençage du transcriptome de l'échantillon n°j et le gène n°i.

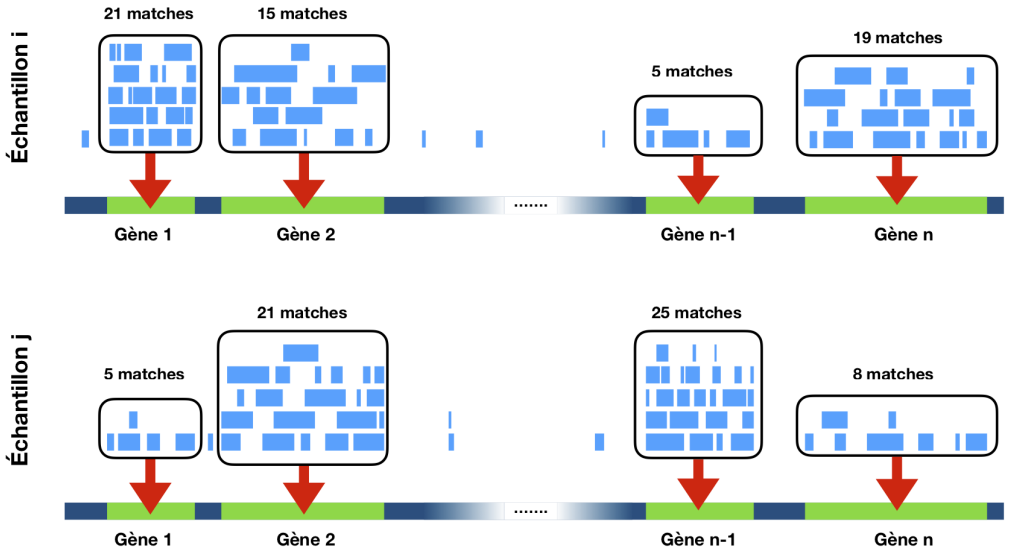

FIGURE 2.5 – Comptage des correspondances avec le génome de référence, pour deux échantillons

2.3.6 Analyse statistique

À partir de la matrice précédente, nous réalisons des analyses statistiques visant à comparer les valeurs des matches par gène entre les échantillons persistants et les contrôles non persistants. On réalise les sorties graphiques désirées, permettant cette comparaison des profils d'expression.

Chapitre 3

Résultats

3.1 Résultats

À l'issu de l'exécution de notre workflow, nous obtenons les deux graphiques suivants :

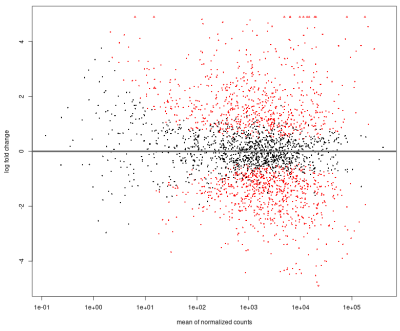

FIGURE 3.1 – MA-plot de l'ensemble du génome de *S. aureus*

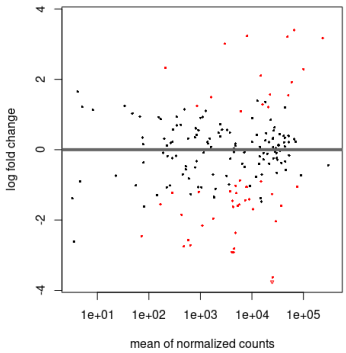

FIGURE 3.2 – MA-plot des gènes de *S. aureus* impliqué traduction

Les MA-plots servent à représenter les différences de profil d'expression entre deux échantillons (les trois échantillons persistants comparés aux trois témoins). Les points rouges correspondent à des gènes dont l'expression est significativement différente chez les bactéries persistantes. Ceux au-dessus de la droite horizontale à y = 0 sont surexprimés chez les persistants, et ceux en-dessous sont sous-exprimés.

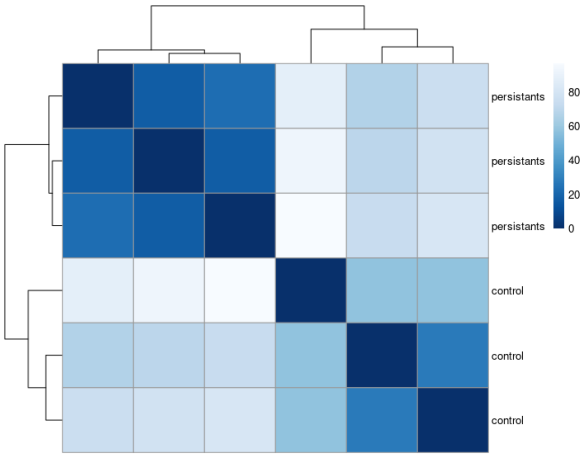

FIGURE 3.3 – Heatmap des distances entre les échantillons, selon leur proximité de transcriptome

Les trois premiers éléments en ligne et colonne correspondent aux échantillons de persistants, et les trois autres aux témoins.

La heatmap permet de visualiser la distance entre les gènes. On constate que les trois échantillons de populations persistantes présentent une plus grande similarité entre eux qu'avec les autres échantillons (les trois témoins), et qu'il en va de même pour les trois témoins. Le clustering en deux groupes souligne cette tendance : les deux groupes obtenus correspondent aux deux types d'échantillons : les persistants d'un côté, les témoins de l'autre.

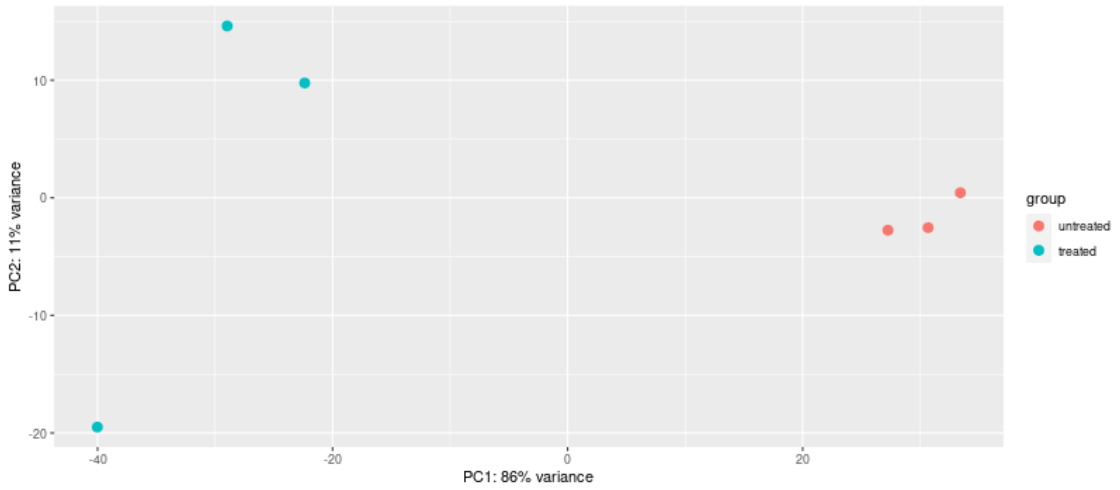

FIGURE 3.4 – Graphe d'ACP des échantillons

Le graphe d'ACP abonde dans ce sens : deux groupes sont clairement séparables selon le premier axe, et ils correspondent bien au deux types d'échantillons (les "untreated" sont les témoins, et les "treated" sont les persistants). Ces deux figures nous permettent d'affirmer que les bactéries persistantes ont un profil d'expression statistiquement différent des non persistantes (témoins).

3.2 Comparaison avec les résultats de l'article

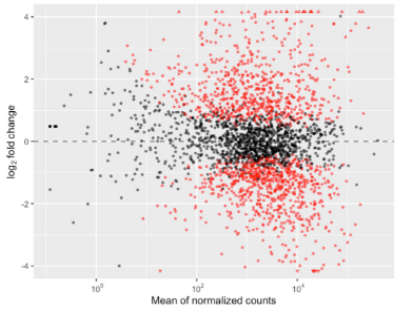

FIGURE 3.5 – MA-plot tous les gènes de l'article

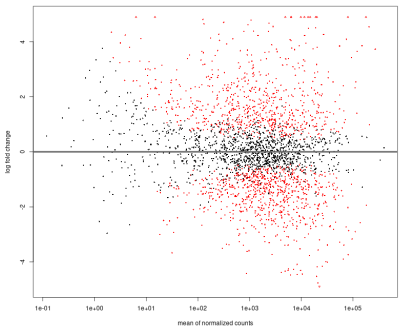

FIGURE 3.6 – MA-plot équivalent obtenu

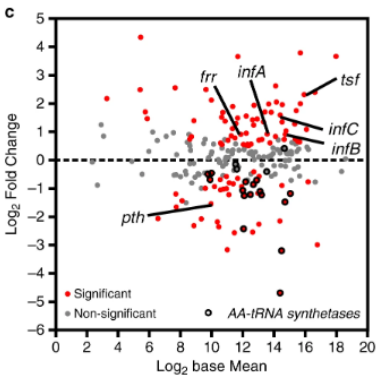

FIGURE 3.7 – MA-plot des gènes de la traduction de l'article

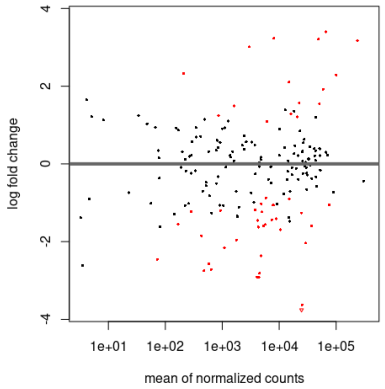

FIGURE 3.8 – MA-plot équivalent obtenu

Visuellement, les deux MA-plots obtenus montrent une allure similaire à ceux de l'article. Néanmoins, nous ne sommes pas parvenus à étiqueter les quelques gènes comme dans l'article pour le MA-plot des gènes impliqués dans la traduction. La comparaison reste donc très limitée : l'absence d'étiquettes ne permet aucune comparaison précise avec l'article pour ces gènes "repères".

3.3 Discussion

Vis à vis de la comparaison mentionnée dans la sous-section précédente, nous pourrions réaliser un graphe de corrélation entre nos graphes et ceux de l'article pour les deux types de MA-plot. En associant ainsi les valeurs de l'article aux nôtres pour chaque gène, nous pourrions estimer leur corrélation.

Nous nous sommes heurtés à certains obstacles lors de la construction d'un flux de travail au plus proche des conditions expérimentales de l'article scientifique. Afin de gagner en efficacité, nous avons développé en parallèle le flux de travail Nextflow et le script R.

L'un des problèmes rencontrés au cours du projet a été la gestion de la RAM des machines virtuelles, et leur capacité de stockage. Les données sont particulièrement lourdes et leur traitement nécessite une grande puissance de calcul, ainsi le manque de mémoire a été une source récurrente d'échec. Pour réduire l'importance de ce problème nous avons travaillé dans un premier temps sur un échantillon de test ne contenant que les mille premières lignes des fichiers .fastq

La conteneurisation a également posé quelques difficultés. Une image docker ne remplira pas sa mission si le logiciel conteneurisé n'est pas installé avec l'ensemble de ses dépendances et avec ses exécutables placés dans le dossier "usr/local/bin". Dans ce contexte, il a été difficile de reproduire l'environnement adéquat pour d'anciennes versions de logiciels car les dépendances, elles aussi anciennes, n'étaient parfois plus disponibles. De

fait nous utilisons d'autres versions des logiciels que celles de l'expérience de l'article, sauf pour le conteneur Bowtie.

Chapitre 4

Conclusion

4.1 Interprétation

La comparaison avec les résultats de l'article scientifique, illustrée par les MA-plots, montre une concordance qualitative entre les deux ensembles de données. Cependant, des obstacles ont été rencontrés lors de l'implémentation du workflow. Dans un premier temps, nous avons surtout été contraints par la gestion de la RAM et de la capacité de stockage des machines virtuelles. La majeure difficulté a consisté à obtenir les mêmes versions des logiciels que celles utilisées dans l'article. Ces contraintes ont conduit à l'utilisation de versions plus récentes pour garantir la reproductibilité des analyses.

Malgré ces défis, l'utilisation de la containerisation, en particulier Docker, a été bénéfique pour assurer la portabilité et la robustesse du flux de travail automatisé. La prise en compte des limitations techniques rencontrées souligne l'importance de l'optimisation des ressources dans les projets de bioinformatique de grande envergure.

En plus des résultats attendus, le graphe d'ACP et la heatmap (cf section 3.2) mettent en évidence la différence significative d'expression génétique des bactéries persistantes.

4.2 Reproductibilité

La reproductibilité dans la recherche scientifique, en particulier dans des domaines tels que la bioinformatique, la biologie computationnelle, peut être entravée par diverses difficultés. Voici les principales difficultés rencontrées lors de notre étude :

— **Gestion des Données** : La taille importante des ensembles de données biologiques, et leur complexité, peuvent rendre difficile la gestion, le partage et la reproduction des analyses. Les machines virtuelles mises à dispositions étaient souvent limitées en capacité de stockages et mettaient un temps considérable à tourner (peut-être aussi considérable à l'échelle industrielle).

— **Dépendances Logicielles** : L'analyse nécessitait l'utilisation de nombreux logiciels tiers, bibliothèques et framework . La gestion des versions de ces dépendances a entraîné des incompatibilités, car l'article, bien que paru récemment, utilise des versions logiciels qui ne sont plus à jour. Il aurait été intéressant de pouvoir utiliser les mêmes versions mais nous n'avons pas pu contourner les problèmes d'incompatibilités. Néanmoins nous avons proposé de faire tourner notre script sur la matrice de comptage de l'article pour observer les résultats qualitativement d'une part avec celles à reproduire et d'autre part avec celles obtenus. Nous pouvons observer les MAplot dans les graphiques qui suivent avec les différentes expressions de gènes :

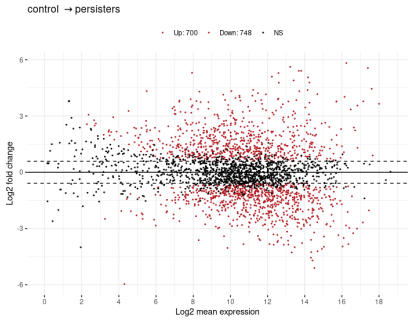

FIGURE 4.1 – MA-plot tous les gènes de l'article

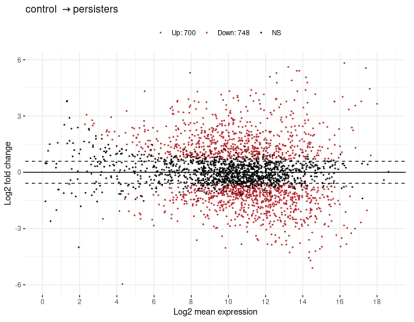

FIGURE 4.2 – MA-plot équivalent obtenu

4.3 Perspectives de la Reproductibilité en Biologie Computationnelle

Les perspectives de la reproductibilité en biologie computationnelle sont cruciales pour assurer la crédibilité, la transparence et l'avancement de la recherche scientifique. Voici quelques-unes des principales perspectives sur la reproductibilité en biologie computationnelle :

1. **Normalisation des Pratiques :** Encourager l'adoption de normes et de bonnes pratiques en matière de gestion de données, de documentation et de programmation. Des initiatives telles que les directives FAIR (Findable, Accessible, Interoperable, Reusable) visent à normaliser la gestion des données scientifiques.
2. **Infrastructure et Outils Dédiés :** Développer des infrastructures et des outils dédiés facilitant la reproductibilité. Des plates-formes de gestion de workflows, des conteneurs logiciels (comme Docker), et des gestionnaires de versions spécifiques aux données peuvent contribuer à cette infrastructure.
3. **Formation et Sensibilisation :** Fournir une formation adéquate aux chercheurs en biologie computationnelle sur les meilleures pratiques de reproductibilité. Cela pourrait inclure l'enseignement de compétences en gestion de données, en documentation, en gestion de versions et en programmation reproductible.
4. **Partage des Données et des Codes :** Encourager le partage systématique des données sources, des codes et des workflows. Les revues scientifiques et les institutions peuvent jouer un rôle en favorisant les pratiques de partage des données en tant que norme.
5. **Évaluation par les Pairs :** Intégrer des évaluations de la reproductibilité dans le processus de révision par les pairs. Les résultats d'une étude devraient être évalués non seulement pour leur validité, mais aussi pour leur reproductibilité.