Université de Rennes
UFR Sciences de la vie et de l'environnement

Développement du package treediff et analyse différentielle de données Hi-C

Gwendaëlle Cardenas

Master 2 - Bio-informatique

Année universitaire : 2022 - 2023

Encadrants : Nathalie Vialaneix et Sylvain Foissac

Tuteur : Vonick Sibut

2 janvier - 30 juin 2023

Structure d'accueil : INRAE
24, Chemin de Borde Rouge, 31320 Auzeville-Tolosane

Université de Rennes

UFR Sciences de la vie & de l'environnement

ENGAGEMENT DE NON PLAGIAT

Je, soussigné (e) Gwendaëlle Cardenas

Etudiant (e) en Master 2 Bio-informatique

Déclare être pleinement informé (e) que le plagiat de documents ou d'une partie de documents publiés sous toute forme de support (y compris l'internet), constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour la rédaction de ce document.

Signature

INSERM U1242 OSS Equipe PROSAC

Centre Eugène Marquis Avenue de la Bataille Flandres Dunkerque
35042 Rennes

Annabelle MONNIER
annabelle.monnier@univ-rennes1.fr

TÉL. 33 (0)2 23 23 61 14

Table des matières

Introduction

Structure tridimensionnelle du génome

Toutes les cellules d'un organisme contiennent la même information génétique. Pourtant, elles ne réalisent pas toutes les mêmes fonctions. L'expression génique est ce qui permet cette différence de fonctionnement. Un changement d'expression peut avoir des conséquences délétères pour l'organisme, comme être l'origine de maladie ou de malformation[1]. En outre, pour le bon fonctionnement de l'organisme, l'expression des gènes est régulée. Cette régulation génique est possible par plusieurs mécanismes, comme la régulation transcriptionnelle, les mécanismes épigénétiques ainsi que la conformation spatiale du génome. Étudier la conformation spatiale du génome peut dévoiler l'origine d'un changement d'expression génique[2]. Dans la suite de ce rapport, c'est ce dernier mécanisme qui va être étudié.

L'ADN est une séquence nucléotidique linéaire qui peut être séparée en chromosomes contenant des gènes entrecoupés de zones intergéniques. Pour être contenu dans une cellule, ce filament génique est condensé, permettant à des régions génomiques éloignées de se rapprocher et d'interagir entre elles. Cette interaction peut influencer l'expression génique. De plus, plus une région de l'ADN est condensée, moins elle est accessible aux mécanismes de lecture du génome, ce qui diminue leur expression.

La condensation de l'ADN n'est pas un phénomène aléatoire. L'organisation spatiale du génome (Figure 1) suit une hiérarchie. Le génome se divise en chromosomes. Dans une cellule, les chromosomes se replient dans leur territoire chromosomique. Des régions du génome qui se replient préférentiellement ensemble, appelés TADs[3] se forment dans les chromosomes, et dans lesquels se trouvent des boucles, qui sont le contact physique entre deux positions génomiques précises. Cette organisation spatiale n'est pas exhaustive, d'autres niveaux d'organisation spatiale peuvent être décrits permettant plus de précisions sur la conformation tridimensionnelle du génome[4].

L'objectif de l'étude de la conformation spatiale est de cartographier le génome afin de comprendre son rôle dans la régulation de l'expression des gènes. Pour étudier la conformation spatiale du génome, différentes méthodes peuvent être utilisées[5]. Les premières sont des méthodes de microscopie comme la méthode Fluorescence in situ hybridization (FISH). Cette méthode permet de visualiser dans la cellule la distance entre deux régions distinctes du génome. Les régions sont ciblées et marquées par fluorescence. La méthode FISH permet d'observer un nombre restreint de régions génomiques à la fois. Grâce aux nouvelles technologies de séquençage d'ADN, des méthodes haut débit peuvent être utilisées comme la méthode High-throughput chromosome conformation capture (Hi-C)[6]. Cette méthode mesure la fréquence des contacts entre paires de régions génomiques.

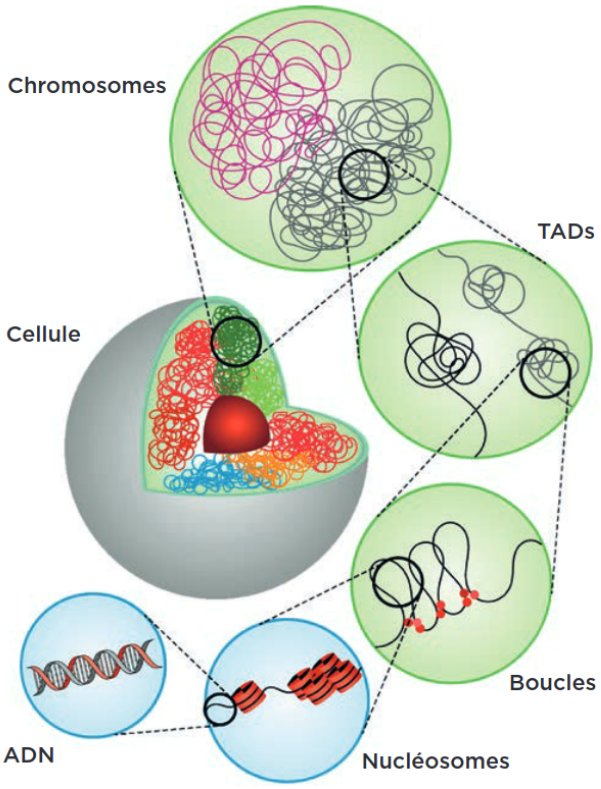

Figure 1 : Organisation spatiale de l'ADN, Biologie, *Comprendre l'organisation spatiale de l'ADN à l'aide de la statistique*, (p.164 - 168) (P. Neuvial, N. Vialaneix, S. Foissac)

La technologie Hi-C

Pour obtenir des données Hi-C (Figure 2), il existe plusieurs protocoles expérimentaux, celui utilisé pour obtenir les données de cette étude est celui de Rao et al, 2014[7]. L'objectif est d'obtenir les paires de fragments d'ADN qui sont spatialement proches, pour cela des bibliothèques sont préparées.

La première étape consiste à fixer le génome avec du formaldéhyde afin de stabiliser les interactions génomiques à identifier. Puis l'ADN est digéré. Les cellules sont traitées par une enzyme de restriction, générant des fragments d'ADN. Les extrémités de ces fragments sont ensuite complétées par la biotine et sont ligaturées par une enzyme ligase T4 DNA, permettant ainsi la création de fragments hybrides composés de deux régions génomiques distinctes mises bout à bout. Le principe de la technique repose sur le fait que la probabilité de ligation entre régions génomique dépend de leur proximité spatiale. Ainsi, plus deux régions génomiques sont proches, plus elles sont susceptibles d'être ligaturées. Après diverses étapes de purification et amplification, ces fragments hybrides sont ensuite séquencés par leurs extrémités. Le séquençage permet de déterminer la séquence d'ADN des extrémités de chaque fragment. La séquence d'une extrémité de fragment est appelée une lecture, chaque fragment produisant une paire de lectures. Après le séquençage, les paires de lectures sont alignées sur la séquence génomique de l'organisme étudié afin de localiser les deux régions génomiques composant chaque fragment hybride. Le nombre de paires de lectures associant deux régions génomiques est comptabilisé.

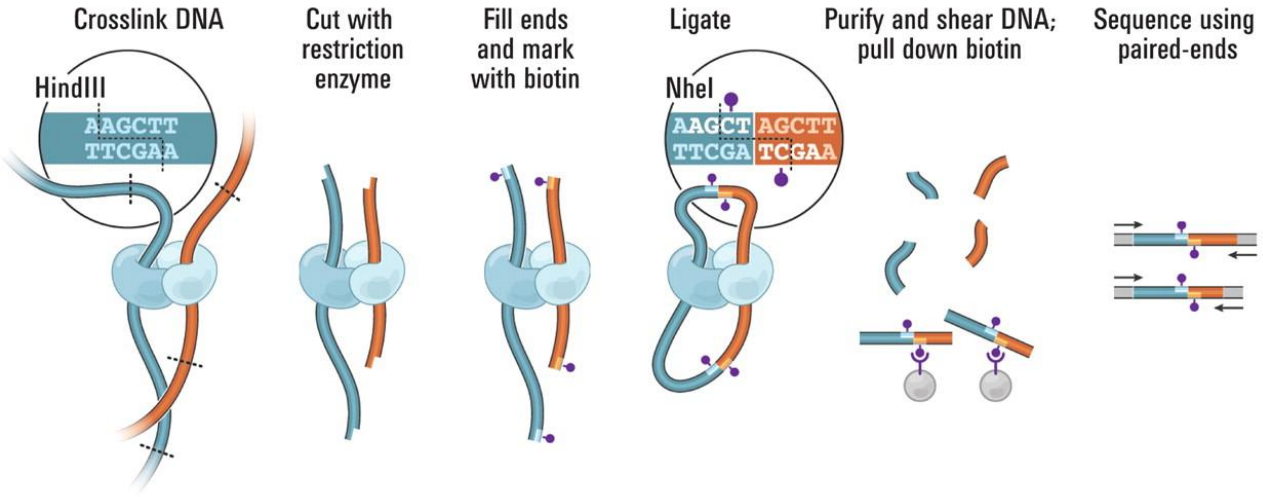

Figure 2 **:** Protocole expérimental de l'Hi-C décrit par Rao et al. 2014[7]

Les comptages obtenus sur un grand nombre de cellules sont ainsi agrégés dans une matrice dans laquelle les lignes et les colonnes correspondent à des régions génomiques de tailles fixes, appelées « bins ». Plus la taille du bin est petite, plus la résolution de la matrice est fine, les tailles de bins variant typiquement de 50kb à 1Mb. En fonction de leur position sur le génome, les lectures d'une paire peuvent être dans le même bin ou non. La matrice de comptage est un dénombrement des paires de lectures se trouvant dans les mêmes bins. Ainsi, chaque entrée de la matrice, appelée interaction, est une valeur numérique : plus cette valeur est grande, plus on a détecté d'interactions entre les régions concernées, ce qui permet d'estimer leur proximité spatiale. Une manière de visualiser les données Hi-C est sous forme de heatmap (Figure 3). Cette manière permet facilement de visualiser des structures de conformation spatiale du génome.

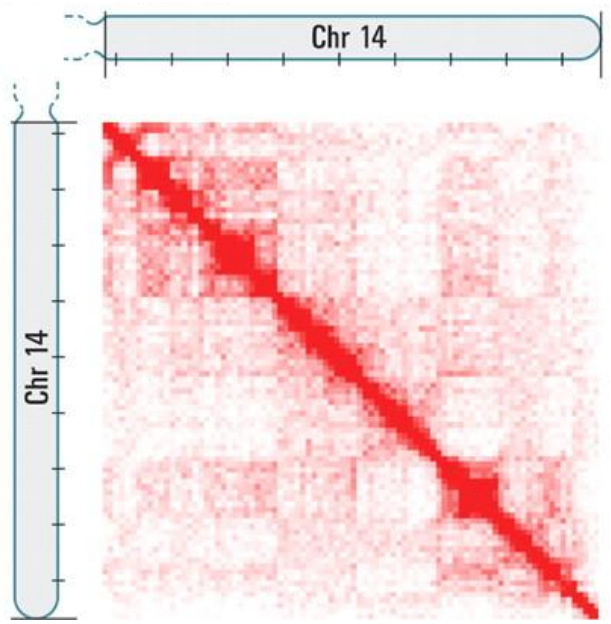

Figure 3 : Heatmap d'un matrice Hi-C d'une partie du chromosome 14 du génome humain (Lieberman-Aiden, 2009[6])

Analyse et comparaison de données Hi-C

L'analyse différentielle des données Hi-C est une étude qui considère deux ensembles de matrices. Chaque ensemble correspond à une condition biologique. Les matrices sont comparées pour identifier des structures d'ADN différentes entre les deux conditions. La méthode la plus simple pour réaliser une analyse différentielle entre deux matrices Hi-C est d'utiliser un Z-score[8]. Pour obtenir la matrice des différences de comptage, chaque valeur de comptage d'une paire de bins (région génomique) de la première matrice est soustrait par la valeur de la même paire de la deuxième matrice. À partir de cette matrice, un Z-score (supposé gaussien centré réduit) est calculé pour chaque paire de bins. Les *p*-valeurs des Z-score sont corrigées et permettent de connaître la significativité des différences entre conditions (ici entre matrices) des paires de bins. Cette méthode est toutefois très restrictive : l'hypothèse gaussienne n'est pas bien adaptée aux données de comptage, d'une part, et la méthode ne permet pas de prendre en compte la variabilité de réplicats qui peuvent avoir été obtenus dans chacune des conditions. Une autre méthode est l'utilisation d'un modèle linéaire généralisé (qui est une généralisation de l'ANOVA à un facteur) en utilisant une loi binomiale négative[9], de manière similaire aux analyses différentielles de données RNA-seq. Ces méthodes permettent de prendre en compte les réplicats des conditions et sont mieux adaptées aux données de comptage mais elles produisent une p-valeur par paire de positions, souvent réparties de manière disparate dans la matrice et n'utilisent pas la notion de hiérarchie de la conformation.

Pour pouvoir prendre en compte l'aspect imbriqué de l'organisation spatiale, une solution peut être de comparer les structures des données en convertissant les matrices en arbres[10,11]. En comparant les topologies des arbres, la hiérarchie de la conformation du génome est prise en compte. Les arbres de clustering hiérarchique, aussi appelés dendrogrammes, sont une représentation simple d'une structure hiérarchique. Dans le cas des données Hi-C, une contrainte d'ordre peut être ajoutée, ne permettant de fusionner le long de la hiérarchie, que des régions juxtaposées. Récemment, une méthode d'analyse de données Hi-C a été mise en place sur cette base : treediff (Neuvial, 2023). La méthode treediff permet de comparer deux ensembles d'arbres, dans lesquels les arbres sont utilisés pour représenter la structure hiérarchique d'une (sous) matrice Hi-C correspondant à un réplicat d'une condition biologique donnée. L'objectif de ce stage est la création d'un package R associé à cette méthode ainsi que, si possible, sa validation sur des données réelles.

Matériel et méthodes

Données utilisées pendant le stage

Durant le stage, deux jeux de données Hi-C sont utilisés. Le premier provient de cellules musculaires de porc et le second de cellules nerveuses de souris. Dans les deux cas, les données se présentent sous la forme d'un ensemble de matrices de comptage divisées en plusieurs groupes que l'on cherche à comparer.

Les données provenant des cellules musculaires de porc ont été utilisées afin de tester, pour la première fois, la méthode treediff. Les premiers tests avaient été effectués avant le début de l'étude. Les cellules proviennent de fœtus de porc de race Large White, de deux stades de gestation différents : 90 et 110 jours[12]. Pour chaque stade de gestation, les cellules musculaires de trois fœtus sont prélevées, ce qui donne trois réplicats par stade. Les rendus des premiers tests sont des scripts R permettant d'effectuer la méthode treediff. Ce sont ces scripts qui ont servi de base pour le stage.

Le second jeu de données utilisé durant le stage pour effectuer une analyse différentielle porte sur des cellules nerveuses issues de souris[13]. La conformation tridimensionnelle du génome entier de trois types de cellules est étudiée, correspondant à divers stades de différenciation cellulaire : les cellules souches embryonnaires (ES), se différencient en deux types de cellules : les cellules des cellules souches neurales (NPC) et en neurones corticaux (CN). La souris comporte 19 chromosomes (hors XY) et il y a 4 réplicats biologiques pour chaque type de cellules. Il y a une matrice de comptage pour chaque réplicats de chaque type de cellules de chaque chromosome, soit 228 matrices. Pour l'analyse, la taille de résolution de la matrice de comptage est de 50 000 pb (il s'agit de la taille d'un bin).

Description de la méthode « treediff »

Représentation de la structure hiérarchique de l'Hi-C avec des arbres

La méthode treediff permet la comparaison de matrices de comptages converties en arbres de clustering hiérarchique (Figure 4). Le clustering hiérarchique est une méthode de classification non supervisée : elle permet de regrouper des éléments en fonction de leur distance. Le résultat de la méthode est sous forme d'arbre et les éléments, appelés feuilles, se retrouvent à la base de l'arbre. Les feuilles sont reliées par des branches dont la longueur est calculée à partir de la distance entre les feuilles. Dans le cas de cette étude, les feuilles sont des bins et on utilise une similarité basée sur les comptages des paires de lectures plutôt qu'une distance.

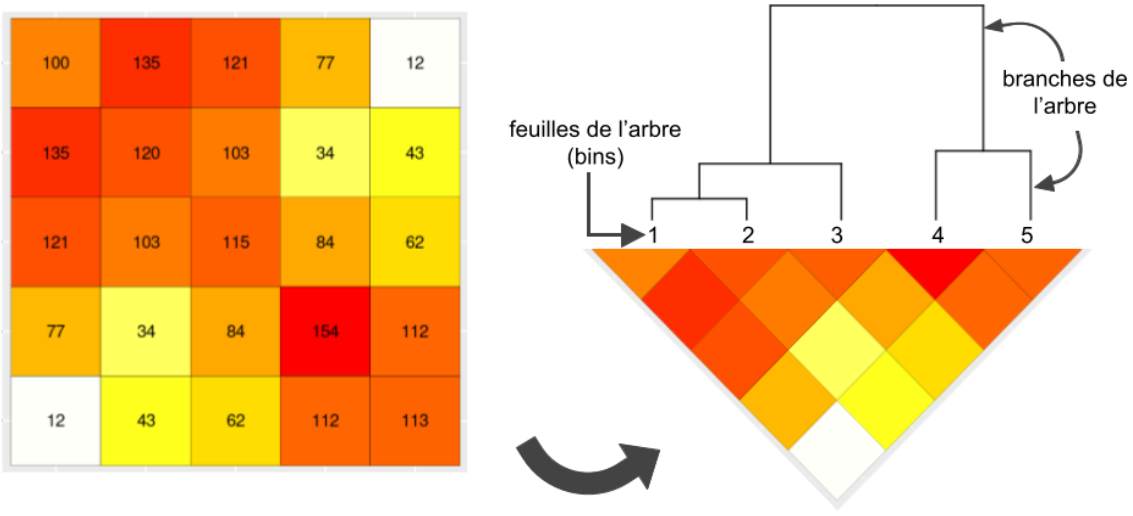

Figure 4 : Représentation d'une matrice Hi-C convertie en arbre de clustering hiérarchique

Ici, le clustering hiérarchique nécessaire pour la conversion des matrices en arbres a été réalisé avec l'outil adjclust[11]. Dans le cas des données Hi-C, une contrainte d'ordre est ajoutée, pour refléter la contiguïté entre régions génomiques adjacentes. En effet, la diagonale de la matrice Hi-C représente le chromosome de manière linéaire. De plus, il y a une corrélation inverse entre la distance entre bins sur le génome et le nombre d'intéractions. Pour ce clustering avec contrainte, seuls les bins adjacents peuvent être fusionnés. De plus, les bins sont fusionnés en utilisant directement la valeur de comptage (comme similarité) plutôt que la distance euclidienne classique. Pour chaque fusion, la différence d'inertie des potentielles fusions est calculée. La fusion choisie est celle avec le plus petit score qui correspond à la plus petite augmentation de l'inertie intra-clusters. Ainsi, les clusters définis sont les plus homogènes possibles (les moins variables). Pour l'étape suivante, les scores sont de nouveau calculés en considérant toutes les fusions adjacentes possibles. De plus, les objets déjà fusionnés sont enlevés. Ces actions sont réitérées jusqu'à ce que tous les objets soient fusionnés ensemble, formant ainsi un arbre. L'algorithme présentant la méthode est décrit dans l'algorithme 1.

1: **Initialisation** : un clustering hiérarchique $A_1 = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$, de dimension n, où chaque cluster est une feuille (c'est à dire, un bin)
2: **for** *t* in 2 to *n-1* **do**
3: Calcul de la différence d'inertie intra-classe après fusion de chaque paire contigüe de $A_t = \{C_1, \ldots C_{n-t+1}\}$, le clustering courant
4: Identification de la différence minimum d'inertie $I_{min} = I(C_k, C_{k+1})$
5: Regroupement de C_k et $C_{k'}$ dans le clustering A_{t+1}
6: **end for**
7: **return** A_n

Algorithme 1 : Classification Ascendante Hiérarchique avec Contrainte d'Ordre

La méthode du « broken-stick »

La méthode du « broken-stick »[14] compare la distribution des comptages d'un arbre consensus des différents réplicats par rapport à une distribution aléatoire. Cette méthode permet de découper les arbres obtenus à partir de chaque chromosome en plusieurs sous-arbres, définissant ainsi des clusters (Figure 5). Un cluster est un groupe de sous-arbres de la même région chromosomique. Ce découpage permet de préciser la comparaison des arbres. Les arbres sont plus petits donc la comparaison est plus localisée. Le découpage est le même pour chaque réplicat, impliquant par exemple que le premier cluster du premier chromosome a les mêmes feuilles pour tous les réplicats. Lors de la comparaison de structure avec la méthode treediff, les groupes des sous-arbres des deux conditions sont comparés. Cette étape permet de cibler la comparaison de structure en régions du chromosome comme illustré dans la figure 5.

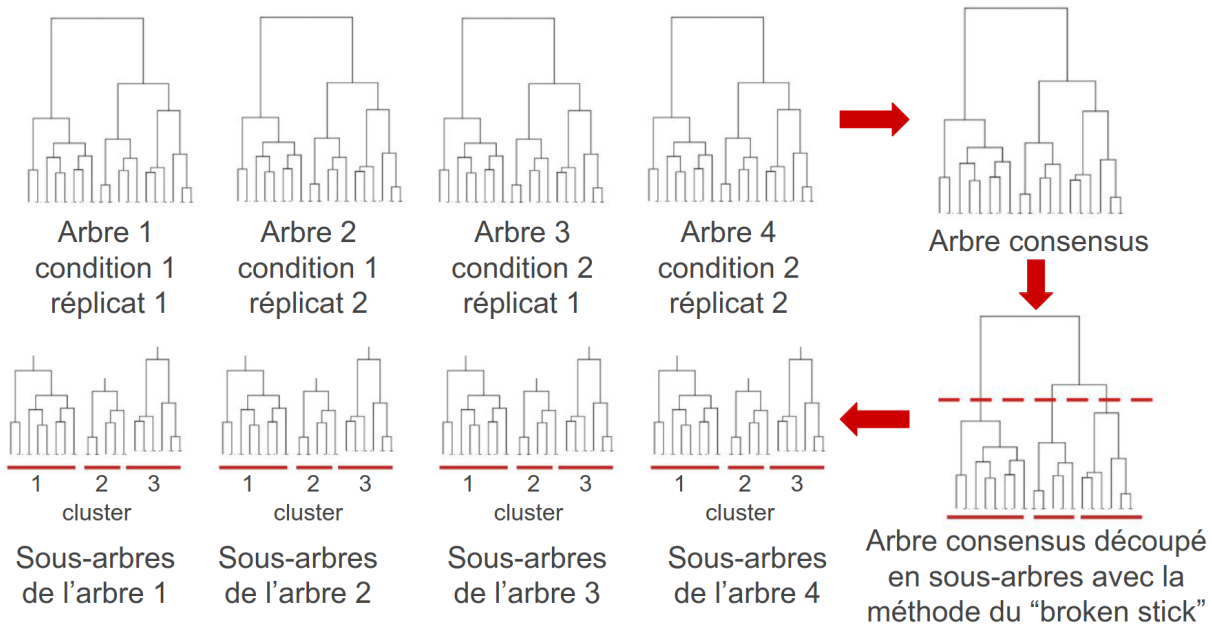

Figure 5 : Schéma représentant la méthode “broken stick”

La méthode treediff

Une fois les arbres (ou sous-arbres) obtenus, la méthode treediff peut être utilisée. Celle-ci se base sur la plus petite distance entre deux feuilles d’un arbre qui est appelée distance cophénétique (Figure 6). Cette distance correspond à la hauteur du nœud le plus haut entre deux paires de feuilles. Les distances cophénétiques de chaque arbre sont mesurées, donc toutes les distances entre paires de feuilles sont mesurées.

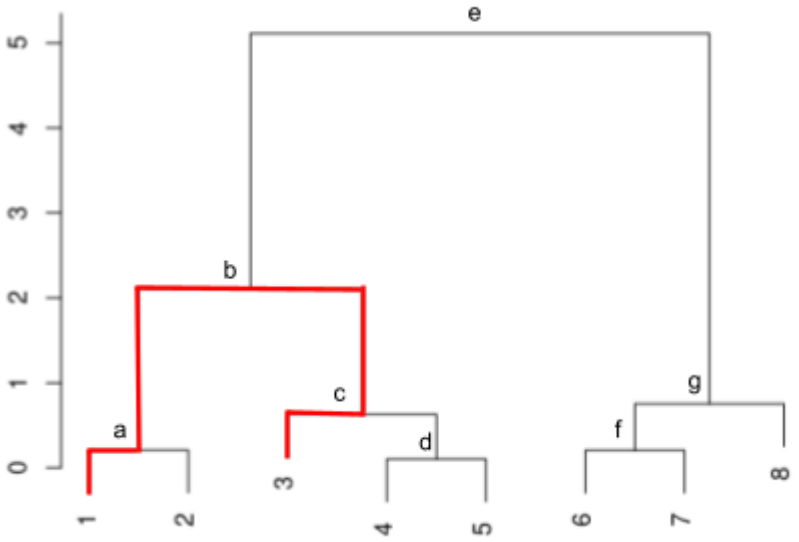

Figure 6 : Arbre de clustering hiérarchique, avec 8 feuilles. La distance cophénétique entre les feuilles 1 et 3 est représentée en rouge et la hauteur du plus haut nœud b est de 2.

Un test statistique comparant les moyennes de deux groupes d'échantillons est utilisé, le test de Student. L'objectif du test est de savoir si les moyennes de distances cophénétiques sont significativement différentes entre les conditions. C’est

pour cela que l'hypothèse nulle (H0) est « les moyennes sont égales » et l'hypothèse alternative (H1) est l'hypothèse réciproque de H0, soit « les moyennes sont différentes ».

Pour chaque paire de feuilles, un test de Student est réalisé à partir de :

$$t_j := \sqrt{\frac{n_1 n_2}{n}} \times \frac{\overline{\mathbf{X}}_j^{(1)} - \overline{\mathbf{X}}_j^{(2)}}{\hat{\sigma}_j},$$

où la statistique de test, t_j avec j = 1,...,p et p le nombre de paires de feuilles, est calculée à partir de $X_j^{(1)}$ moyennes des distances cophénétiques de la première condition de la paire j et $X_j^{(2)}$ pour la seconde condition à la paire j, de σ_j l'estimateur global de l'écart type des distances cophénétiques (« pooling » des écarts types empiriques des deux conditions), de n_1 le nombre de réplicats de la première condition biologique, n_2 le nombre de réplicats de la seconde et $n = n_1 + n_2$.

Dans la méthode treediff, les variances utilisées sont des variances « modérées »[15], qui sont ré-estimées globalement par une approche bayésienne pour en diminuer leur impact sur le résultat du test. Il y a une régulation de la variance due à la multitude de tests effectués. Il s'agit d'une régulation bayésienne empirique. Les variances sont re-calculées à partir d'une distribution de khi² et en tenant compte de toutes les variances initiales. Cette méthode est implémentée dans la fonction squeezeVar du package R limma, et est couramment utilisée pour l'analyse différentielle de données transcriptomiques.

Pour chaque statistique obtenue (une par paire de feuilles), une p-valeur π_j est calculée.

$$\pi_j = 2(1 - F_{\nu_0+n-2}(|t_j|))$$

où F_{v0} est la fonction de répartition de la distribution de Student, t_j la statistique et n le nombre de degrés de liberté, $n = v_0 + n_1 + n_2$ avec v_0 un paramètre issu de l'étape de modération des variances.

Le test de Student permet d'identifier des différences entre paires de feuilles de conditions différentes mais pas entre topologies des arbres. Pour comparer les groupes d'arbres à partir des p-valeurs obtenues, une agrégation de p-valeurs est effectuée qui correspond à tester l'union des hypothèses H_0 individuelles au sein d'un arbre.

Pour ce faire, la méthode de Simes[16] est utilisée. Cette méthode permet de calculer une p-valeur globale pour chaque comparaison de clusters. Pour obtenir la p-valeur de l'agrégation (π_{Simes}), les p-valeurs correspondant à toutes les paires de bins du cluster sont rangées par ordre croissant, multipliées par le nombre de tests (correspondant au nombre de paires de feuilles) et divisées chacune par leur rang. La plus petite p-valeur est donc finalement multipliée par le nombre de tests alors que la plus grande reste finalement inchangée. Parmi tous les quotients obtenus, celui avec le plus petit résultat est la p-valeur finale retenue.

$$\pi_{\text{Simes}} := \min \left\{ p\frac{\pi_{(j)}}{j},\ j = 1, \ldots, p \right\}$$

où π_j est la *p*-valeur à la paire de feuilles *j* et *p* est le nombre de paires de feuilles.

Le résultat de l'agrégation est une *p*-valeur unique pour un ensemble de sous-arbres (cluster) de deux conditions différentes testant donc l'hypothèse nulle d'identité entre sous-arbres. Aussi, avec une erreur α de 5%, l'hypothèse nulle est rejetée lorsque la *p*-valeur obtenue est inférieure à 0,05. Les sous-arbres sont alors déclarés significativement différents, ce qui signifie que la conformation spatiale de la région génomique correspondante est significativement différente entre les deux conditions biologiques.

Correction des p-valeurs

La correction de Benjamini-Hochberg[17] est une méthode utilisée pour contrôler le taux de fausses découvertes ou FDR (False Discovery Rate) lors de l'analyse de multiples tests d'hypothèses simultanés. Une correction de Benjamini-Hochberg est réalisée pour chaque p-valeurs de chaque cluster. Pour commencer les p-valeurs sont classées par ordre croissant. Puis une nouvelle p-valeur est attribuée, pour la calculer, le quotient du rang de la p-valeur et du nombre de p-valeurs total est multiplié par la p-valeur initiale. Le seuillage des p-valeurs ajustées à 5% permet de contrôler le taux de fausses découvertes au niveau de 5% sur l'ensemble des tests déclarés positifs.

Résultats

Cette partie présente les résultats obtenus durant le stage. La première partie porte sur le package treediff. Elle explique la manière dont j'ai implémenté le package et comment l'utiliser. La seconde partie est l'application de la méthode treediff pour la réalisation d'une analyse différentielle, sur des données Hi-C, via le package.

Implémentation de la méthode : le package treediff

La méthode développée antérieurement a été implémentée en R, dans le package treediff durant le stage et rendue disponible sur le CRAN (https://cran.r-project.org/package=treediff).

Le support de départ, pour la création du package, est un ensemble de scripts en R qui testent la méthode treediff. À partir de matrices de comptages Hi-C, les scripts avaient été développés et utilisés pour réaliser l'analyse différentielle des données Hi-C provenant de muscles de porc de deux stades de gestation 90 et 110 jours[12]. Ces scripts étant spécifiques aux données de l'analyse réalisée, il n'était pas possible de transposer la méthode à d'autres données Hi-C. L'objectif du stage était donc, en se servant des scripts, d'implémenter un package R qui permette l'analyse différentielle de tous les types de données Hi-C. Le package treediff permet de réaliser une comparaison de matrices de deux conditions différentes. Le package

permet de partir des matrices de comptages brutes et donne un résultat sous forme de tests statistiques. Pour cela, deux versions du package ont été implémentées.

J'ai ainsi réalisé une première version du package qui ne contenait que la fonction treediff comparant deux ensembles d'arbres. J'ai implémenté cette fonction sous forme de test statistique donnant une p-valeur pour un ensemble d'arbre comparée. Pour cela, j'ai rassemblé dans une fonction les différentes étapes de la méthode treediff contenue dans les scripts de départs. Cela permet de dissocier l'utilisation de la méthode treediff et la création des arbres. En effet, dans les scripts de départ, les étapes n'étant pas regroupées, pour obtenir le résultat de la comparaison d'autres étapes devaient être réalisées. De plus, les données de départ peuvent être tout type d'arbres, des arbres de clustering hiérarchique issue de données Hi-C mais aussi, par exemple, des données obtenues sous forme d'arbres comme les données phylogéniques. Dans le cas des données Hi-C, cette fonction permet de réaliser l'analyse différentielle. Pour réaliser l'analyse différentielle des données Hi-C, chaque ensemble représente une condition biologique avec ses réplicats biologiques.

Puis j'ai implémenté une seconde version du package qui est composée de 5 fonctions. Les fonctions peuvent être utilisées indépendamment les unes des autres (Figure 7) :

- HiCDOCDataSet : importe et transforme les données Hi-C en object HiCDOCDataSet
- normalizeCount : normalise les données de comptages
- clusterTree : crée des sous-matrices et les transforme en arbre
- treediff : compare deux ensembles d'arbres (déjà dans la première version)
- HiC2Tree : reprend toutes les fonctions et réalise une analyse des données Hi-C allant de l'importation de données à l'analyse différentielle

La fonction principale HiC2Tree prend en entrée les fichiers des matrices de comptages brutes et réalise l'analyse différentielle. Cette fonction est composée de quatre fonctions. Les fonctions sont implémentées afin de pouvoir être utilisées les unes après les autres, cela permet à l'utilisateur de pouvoir réaliser seulement certaines étapes du processus ou de vérifier chaque étape du processus d'analyse. Dans notre étude, les scripts de l'analyse différentielle des données de porc servent toujours de base pour la création des fonctions.

La première fonction permet de prendre en compte les données Hi-C d'autres expérimentations et pouvant avoir un format de données Hi-C différent. La fonction que j'ai implémentée pour cette étape est HiCDOCDataSet. La fonction permet la conversion de quatre formats possibles de données Hi-C en objet de classe HiCDOCDataSet par chromosome contenant tous les réplicats. Cette fonction permet de rendre réutilisable le code pour d'autres données, ce qui n'était pas présent dans les scripts de départ. Pour cela, le package R HiCDOC est utilisé. La classe HiCDOCDataSet est héritée de la classe standard des données Hi-C sur R InteractionSet. Ce format permet de faciliter les analyses sur les données Hi-C.

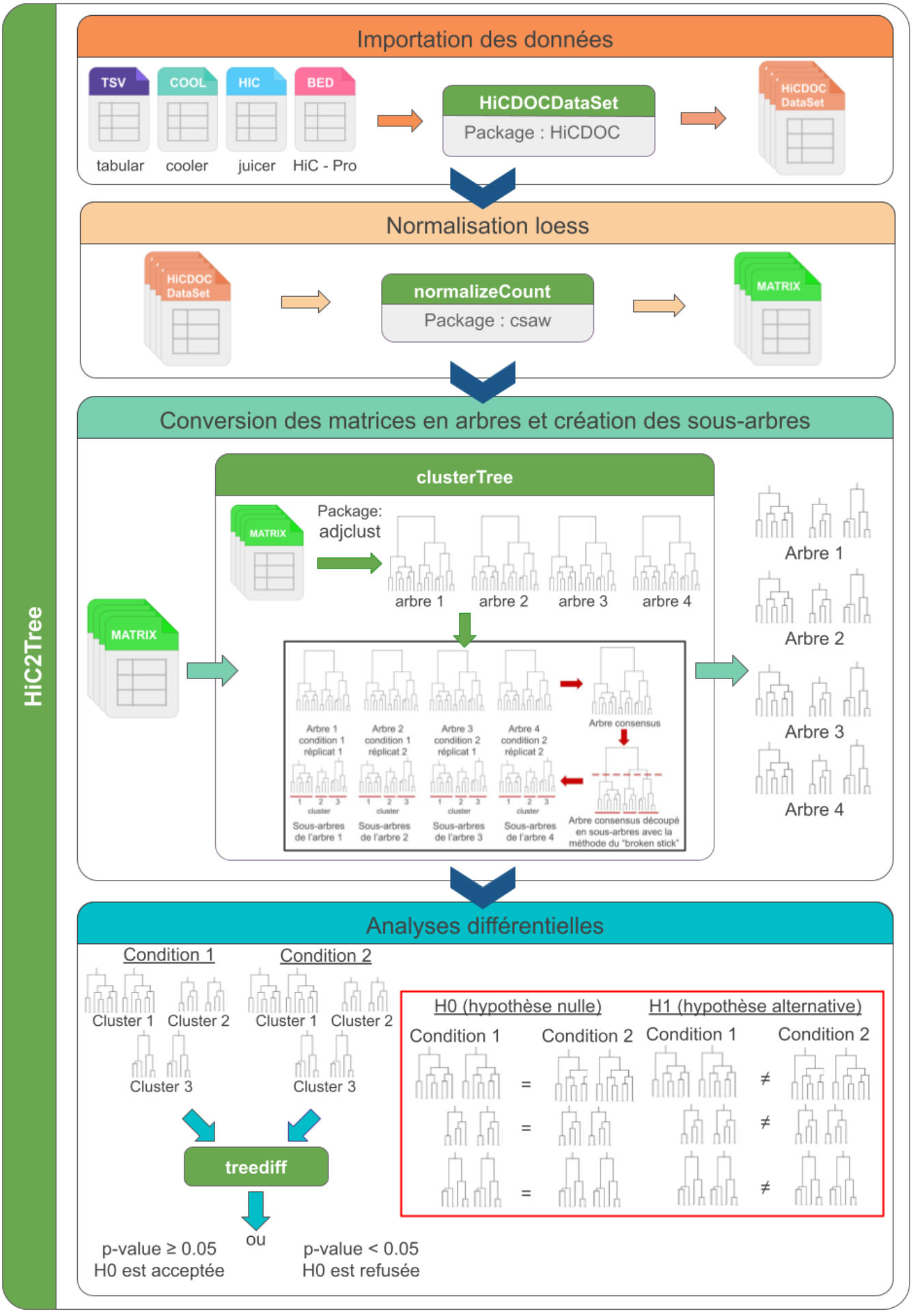

Figure 7 : Workflow du package treediff

La deuxième fonction normalise des matrices de comptage. Une normalisation loess est effectuée sur les matrices de comptage. La normalisation permet de comparer les matrices entre elles, en excluant le biais de profondeur. La deuxième fonction normalizeCount réalise cette normalisation. La fonction peut prendre en entrée des matrices de comptages bruts ou des objets HiCDOCDataSet obtenus par la fonction précédente. La fonction normOffsets du package R csaw est utilisée et légèrement modifiée pour effectuer la normalisation. Les scripts de départ réalisent une normalisation sur une matrice de comptages brute. J'ai modifié le code afin de pouvoir réaliser la normalisation sur plusieurs matrices à la fois, et ajouté la possibilité d'utiliser directement le format HiCDOCDataSet avec une implémentation utilisant le typage en classe S3 de R.

La troisième fonction clusterTree regroupe deux étapes : le découpage des chromosomes en clusters et la conversion des matrices en arbres. Pour ce faire, j'ai implémenté une fonction interne du package permet, à partir d'une matrice de comptage d'un chromosome, de le découper. La méthode utilisée pour découper le chromosome est « broken stick » telle qu'implémentée dans le package R adjclust. Cette méthode ne s'utilise pas directement sur des matrices de comptage mais sur des arbres. Les matrices de comptage correspondant à tous les réplicats d'un même chromosome sont fusionnées (somme) puis converties en un arbre consensus en utilisant un clustering hiérarchique avec une contrainte d'ordre, en utilisant la fonction adjClust du package R adjclust. Chaque matrice de réplicats est découpée en plusieurs sous-matrices en suivant le découpage de l'arbre consensus. J'ai ensuite implémenté la seconde fonction interne du package qui convertit les sous sous-matrices en arbres, toujours en utilisant la fonction adjClust. Pour l'implémentation de cette fonction, j'ai utilisé différentes parties des scripts de départ et je l'ai rassemblées rassemblées dans les deux sous-fonctions. Puis j'ai utilisé ces fonctions pour implémenter clusterTree afin de pouvoir prendre en entrée les données de sortie de normalizeCount.

La dernière fonction de HiC2Tree est treediff, qui était déjà disponible dans la première version du package.

Pour réaliser l'implémentation de HiC2Tree, j'ai utilisé les quatres fonctions décrites précédemment. Les entrées et les sorties des différentes fonctions ont été testées par rapport aux résultats obtenus avant le stage afin d'obtenir le résultat escompté.

Avant d'être soumis au CRAN, pour chaque fonction externe du package j'ai ajouté des tests fonctionnels. Ils permettent de vérifier ce qui est utilisé en entrée de chaque fonction et délivrent un message d'erreur en cas de mauvais renseignement. Pour chaque fonction j'ai ajouté des tests unitaires, ce qui permet de valider la fonctionnalité du code (ces tests sont automatiquement exécutés lors de la soumission du package sur le CRAN). Les tests unitaires sont réalisés avec le package R testthat[18]. J'ai également rédigé toute la documentation, en anglais, en utilisant le format roxygen du package R roxygen2[19], pour permettre l'utilisation des fonctions par d'autres utilisateurs.

La dernière étape avant la soumission au CRAN est de relire et de modifier le code afin que l'implémentation convienne au standard du CRAN. Pour réaliser cette étape, j'ai utilisé la fonction check du package R devtools[20] avec l'option *--as-cran* qui effectue les mêmes tests que ceux utilisés lors de la soumission. Une fois cette étape réalisée, le package a été soumis au CRAN.

Application du package treediff

Avec le package treediff, j'ai réalisé une analyse de données réelles provenant d'une expérience Hi-C menée à grande échelle[13]. Pour la réaliser, les données des différents types de cellules nerveuses issues de la souris, sont utilisées. Les rendu s de cette analyse sont des fichiers html (obtenus à partir d'un rapport Rmarkdown), contenant les différents résultats des étapes du package treediff ainsi qu'une analyse différentielle des p-valeurs obtenue.

Pour cette étude, trois comparaisons de cellules ont été réalisées : les cellules CN contre les cellules NPC, les cellules ES contre les cellules CN et la dernière ES contre NPC. Pour chaque comparaison, l'analyse a été réalisée avec le package treediff. Les p-valeurs obtenues sont ensuite corrigées, par chromosomes, avec la procédure de Benjamini-Hochberg[17]. Cette étape de correction est réalisée après l'utilisation du package treediff.

L'objectif premier est de tester la méthode treediff avec des données réelles et à grande échelle. Cette analyse permet donc de regarder si des biais existent. Le test permet aussi de savoir combien de temps dure une analyse avec un grand nombre de données. Ce temps est un critère dans le choix d'utilisation de la méthode. En effet, une méthode très fiable mais prenant beaucoup de temps à être exécutée peut être un frein pour son utilisation.

Les différents scripts sont soumis sur la plateforme genotoul. Pour réaliser l'analyse des données d'une comparaison du génome entier de deux types cellulaires de souris, il faut compter environ trois heures et les informations stockées prennent un peu plus de 3Go. La fonction qui met le plus de temps à s'exécuter (environ deux heures) est la fonction de normalisation et plus précisément, l'assemblage des matrices de comptage par chromosomes pour les objets HiCDOCDataSet. Les fonctions HiDOCDataset s'exécutent en quelques minutes et clusterTree en quelques dizaines de minutes. Pour ce qui est de la fonction treediff, quelques secondes suffisent.

Analyses des données souris

Le second objectif de l'analyse des données est l'obtention des résultats. En effet, ces données n'avaient jamais été analysées, en prenant en compte l'organisation hiérarchique du génome. Pour cela les fonctions du package treediff sont utilisées. Pour réaliser cette analyse avec le package, il y a deux possibilités : la première, utilisant la fonction HiC2Tree qui réalise toutes les étapes de l'analyse jusqu'à l'obtention des p-valeurs, ou la seconde, en réalisant chaque étape de l'analyse séparément. Pour cette analyse, la seconde option est utilisée, cela permet d'obtenir toutes les informations de chaque étape.

La fonction HiCDOCDataSet est utilisée en premier. Les fichiers d'entrées sont des fichiers du format HiC-Pro, il y a 228 fichiers d'entrées, un par matrice Hi-C. Avec cette première fonction, les fichiers sont tous convertis en objets HiCDOCDataSet, un objet par matrice.

Normalisation loess

La deuxième étape est la normalisation. Pour normaliser les matrices, la fonction normalizeCount est utilisée. Grâce à cette fonction, une matrice normalisée est obtenue par chromosome, il y a donc 19 matrices de comptages normalisées pour chaque comparaison cellulaire.

	nombre de comptes	minimum	maximum	moyenne	mediane	1er quartile	3ème quartile
Données brutes							
ES vs CN	341011727	1.000	28238.00	12.440	3.000	1.000	5.000
ES vs NPC	376232596	1.000	28238.00	14.810	3.000	2.000	7.000
CN vs NPC	371624107	1.000	19452.00	12.490	3.000	1.000	6.000
Données normalisées							
ES vs CN	341011727	0.280	12203.03	10.610	2.378	1.274	4.610
ES vs NPC	376232596	0.253	15220.72	12.674	2.958	1.359	6.038
CN vs NPC	371624107	0.237	11203.38	11.127	2.514	1.482	4.965

Table 1 : Résumé du résultat des normalisations des trois comparaisons de type cellulaire

La Table 1 agrège différentes informations sur les données de comptage avant et après normalisation. On observe que le nombre de paires de bins avec un comptage supérieur à 0 est le même pour les données normalisées et brutes, soit entre 341 011 727 et 376 232 596. La comparaison qui recense le plus de comptages est ES vs NPC et celle qui en recense le moins est ES vs CN. On peut en déduire que le type cellulaire avec le plus de comptage est NPC suivi de ES et pour finir CN. Le nombre de comptage reflète la profondeur de séquençage et donc le nombre de lectures séquencées par conditions.

La comparaison ES vs NPC a le plus de comptages et la valeur moyenne de comptage la plus grande pour les données brutes. On peut supposer qu'il y a bien un léger biais dû au nombre de comptages. Après normalisation, ce phénomène est toujours observé, mais les écarts des moyennes sont plus faibles. La différence entre les valeurs moyennes extrêmes est 2,37 pour les données brutes et de 2,064 pour les données normalisées. Il y a donc une diminution de l'écart des moyennes. La normalisation semble gommer ce biais de comptage. On peut aussi observer que les valeurs après normalisation sont plus faibles. La normalisation minimise les valeurs extrêmes et, dans le cas des matrices de comptages, les grandes valeurs.

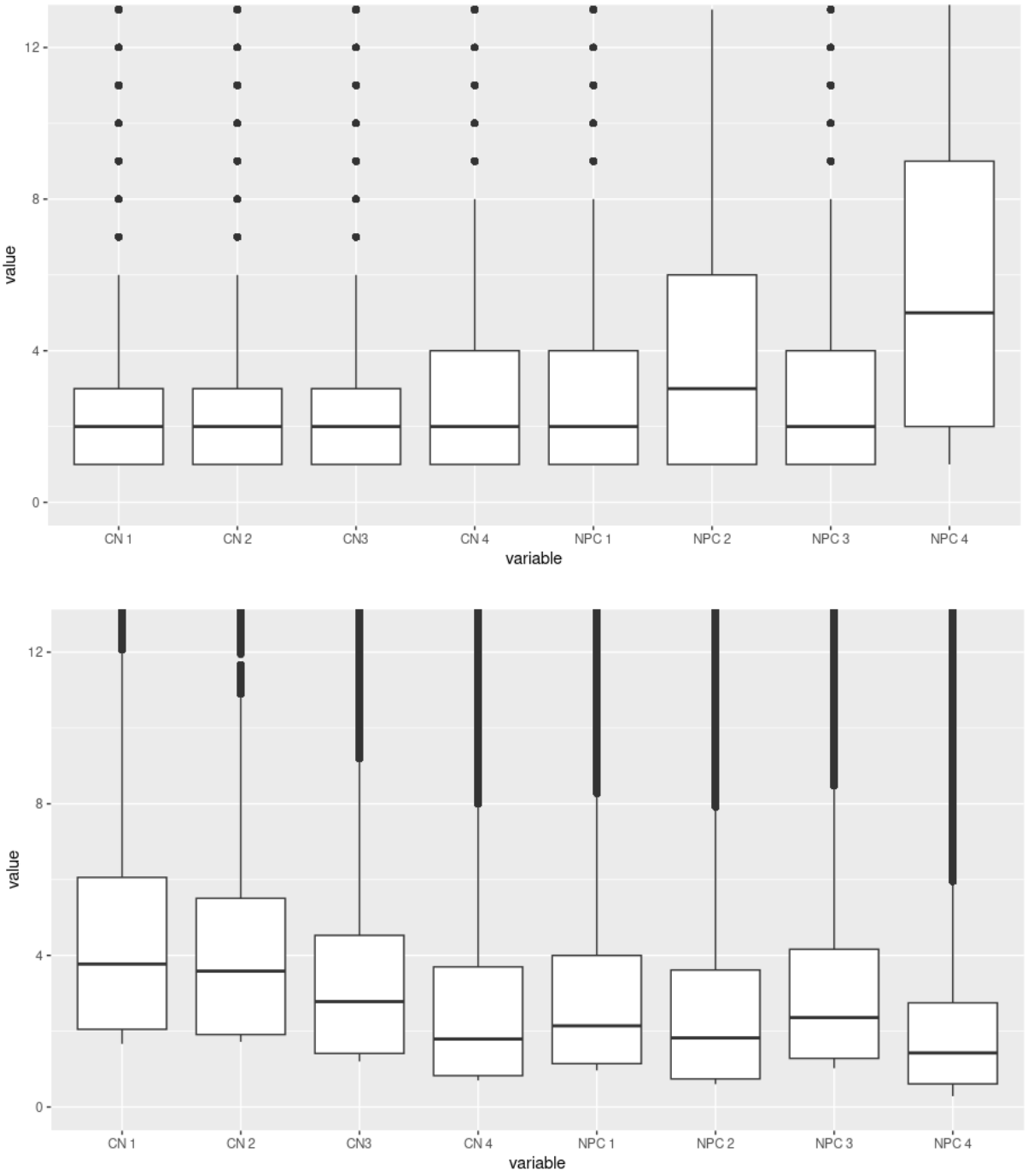

Figure 8: Distribution des comptages du chromosome 1 de la comparaison cellulaire CN vs NCP. Données de comptage bruts (en haut) et données normalisées (en bas).

La Figure 8 montre la distribution des données avant et après normalisation sous forme de boîte à moustaches. Cette figure a été réalisée avec les données du chromosome 1 et de la comparaison CN vs NPC. Les quatre premières boîtes à moustaches sont les réplicats du type cellulaire CN et les quatre dernières du type cellulaire NPC. On remarque que les réplicats 2 et 4 de NPC ont des valeurs de comptage plus importantes que les autres réplicats. Leurs moyennes sont plus hautes et les valeurs sont plus dispersées. Après normalisation, on remarque que

cela n'est plus le cas pour les deux réplicats. En effet, leur moyenne a baissé et leur valeur est moins dispersée.

Obtention des arbres

Les matrices normalisées sont ensuite converties en arbre puis découpées en sous-arbres par la fonction clusterTree. La Table 2 montre le découpage global des arbres des trois comparaisons de types cellulaires.

	ES vs CN	ES vs NPC	CN vs NCP
Nombre de clusters	1877	1877	1757
Nombre de sous-arbres	15016	15016	14056

Table 2 : Table du nombre de cluster et de sous-arbre pour les trois comparaisons de types cellulaires

D'après la Table 2, les comparaisons ES vs CN et ES vs NPC ont le même nombre de clusters et de sous-arbres, soit 1877 clusters et 15016 sous-arbres chacun. Les deux premières comparaisons ont en commun le type cellulaire ES. La troisième comparaison CN vs NPC est différente des deux premières et compte un peu moins de clusters (1757) et de sous-arbres (14056). Ces résultats sont ensuite détaillés par chromosomes dans la Table 3.

	ES vs CN		ES vs NPC		CN vs NPC	
chromosome	sous-arbres	clusters	sous-arbres	clusters	sous-arbres	clusters
1	1312	164	1280	160	1184	148
2	1256	157	1240	155	1112	139
3	1000	125	1032	129	872	109
4	784	98	808	101	728	91
5	888	111	784	98	832	104
6	848	106	840	105	784	98
7	800	100	800	100	808	101
8	688	86	688	86	632	79
9	928	116	976	122	816	102
10	760	95	784	98	752	94
11	832	104	840	105	856	107
12	472	59	472	59	416	52
13	632	79	640	80	624	78
14	624	78	608	76	632	79
15	760	95	744	93	688	86
16	704	88	736	92	648	81
17	576	72	536	67	632	79
18	632	79	672	84	552	69
19	520	65	536	67	488	61

Table 3 : Table du nombre d'arbres et de clusters des trois comparaisons de types cellulaires pour les 19 chromosomes de la souris.

La Table 2 montre que ES vs CN et ES vs NPC ont le même nombre de clusters et de sous-arbres, il aurait pu être attendu que le nombre de clusters par chromosome soit le même pour les deux comparaisons mais cela n'est pas le cas. En effet, la Table 3 montre que le nombre de clusters est différent pour 16 chromosomes néanmoins, le nombre de clusters reste proche. La différence de cluster est entre 0 et 5 pour 17 chromosomes. Le chromosome 9 compte 6 clusters de différences et le chromosome 5, 13 clusters de différences. Pour la comparaison CN vs NPC, il y a en général moins de clusters par chromosomes par rapport aux autres comparaisons, c'est le cas pour 13 chromosomes.

On remarque que le nombre de clusters semble diminuer globalement avec l'ordre des chromosomes. La numérotation des chromosomes se fait souvent en fonction de leur taille. Le chromosome 1 est celui qui a le plus de clusters pour chaque comparaisons, entre 164 et 148, cela semble cohérent car il s'agit du plus grand chromosome, à l'inverse le chromosome 19 est le plus petit et compte moins de cluster, entre 67 et 61, mais il n'est que le deuxième chromosome avec le moins de cluster. En effet, le chromosome 12 est celui qui en compte le moins entre 59 et 52.

Analyse différentielle

L'analyse différentielle est réalisée pour les trois comparaisons, une p-valeur est obtenue pour chaque cluster puis les p-valeurs sont ajustées avec la correction de Benjamini-Hochberg.

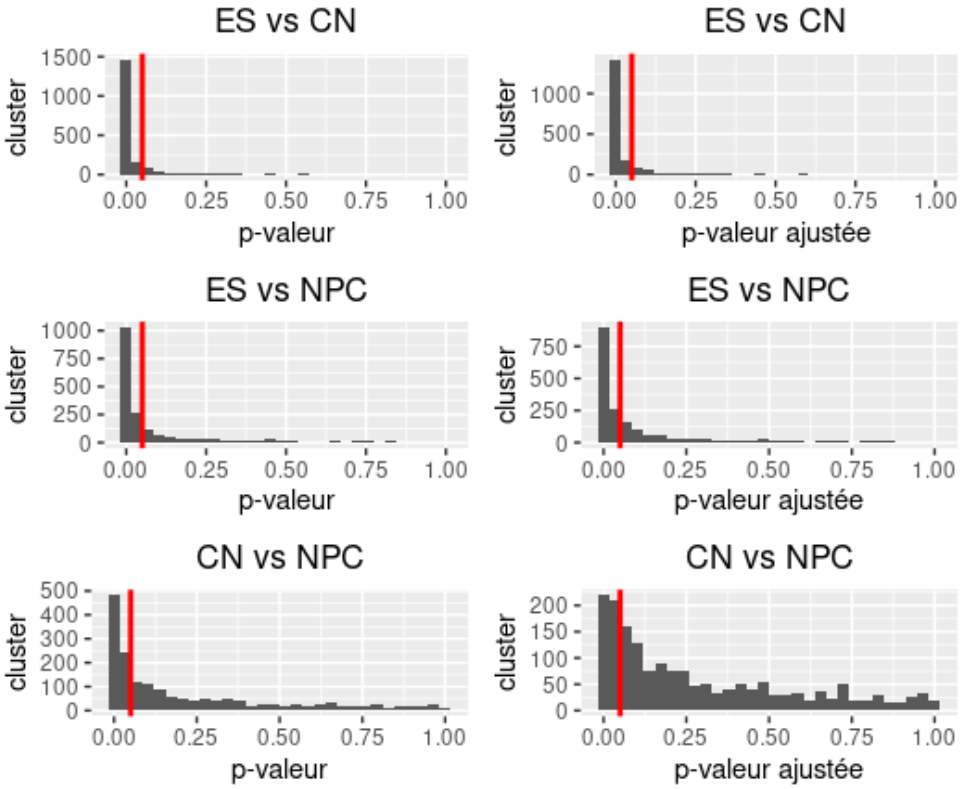

Figure 9: Répartition des cluster à partir des p-valeurs et des p-valeurs ajustées

La Figure 9 montre que la correction effectuée sur les p-valeurs les augmente. Cette augmentation est peu visible sur les deux premières comparaisons mais bien visible sur la dernière. Cette augmentation peut aussi être observée avec la Table 3.

	Clusters différentiels	Clusters différentiels après BH
ES vs CN	1612 (85.9%)	1596 (85%)
ES vs NPC	1279 (68.1%)	1149 (61.2%)
CN vs NPC	717 (40.8%)	423 (24.1%)

Table 4 : Table du nombre de cluster différentiels pour les trois comparaisons de type cellulaire avec et après correction de la p-valeur

La Table 4 permet d'observer que le nombre de clusters différentiels diminue après la correction. Pour les deux premières comparaisons, le nombre diminue de 0.9% et de 6.9% alors que pour la dernière, le nombre diminue de 16.7%. La table permet d'appuyer ce qui est visible sur les graphiques. La baisse du nombre de clusters différentiels est due à la correction des p-valeurs. La correction à tendance à augmenter les p-valeurs ce qui permet à certaines p-valeurs de dépasser le seuil de 0.05 et une fois dépassée le cluster associé à la p-valeur n'est plus considéré comme différentiel.

Cette table permet, aussi, de remarquer que la comparaison avec le plus de clusters différentiels est ES vs CN avec 85% de clusters différentiels suivi de la comparaison ES vs NPC avec 61.2% de clusters différentiels et pour terminer la comparaison avec le moins de clusters différentiels est CN vs NCP avec 24.1%. Les types cellulaires qui semblent les plus proches sont CN et NCP car le nombre de clusters différentiels est le moins élevé. À l'inverse, les types cellulaires qui semblent les plus éloignés sont ES et CN, avec le nombre de clusters différentiels le plus grand. La Table 5 permet de voir le nombre de clusters différentiels par chromosomes.

	ES vs CN		ES vs NPC		CN vs NPC	
chromosome	clusters différentiels (en %)	clusters différentiels BH (en %)	clusters différentiels (en %)	clusters différentiels BH (en %)	clusters différentiels (en %)	clusters différentiels BH (en %)
1	84.1	81.7	64.4	59.4	40.5	29.7
2	87.9	87.3	70.3	58.7	40.3	30.2
3	84.0	84.0	65.1	56.6	44.0	22.0
4	88.8	87.8	72.3	65.3	49.5	33.0
5	85.6	84.7	73.5	68.4	45.2	25.0
6	91.5	91.5	65.7	55.2	45.9	27.6
7	85.0	85.0	66.0	60.0	49.5	36.6
8	86.0	83.7	76.7	73.3	48.1	35.4
9	84.5	82.8	63.1	54.9	44.1	25.5
10	83.2	81.1	76.5	69.4	38.3	19.1
11	91.3	91.3	70.5	63.8	34.6	15.0
12	100.0	100.0	88.1	88.1	53.8	38.5
13	91.1	91.1	73.8	65.0	35.9	21.8
14	84.6	83.3	71.1	65.8	32.9	20.3
15	78.9	75.8	50.5	40.9	32.6	14.0
16	84.1	83.0	60.9	50.0	30.9	18.5
17	81.9	80.6	73.1	70.1	41.8	32.9
18	83.5	79.7	61.9	54.8	29.0	8.7
19	76.9	76.9	62.7	56.7	36.1	11.5

Table 5 : Table du nombre de cluster différentiels de chaque chromosome pour les trois comparaisons de type cellulaire avec et après correction de la p-valeur

Sur la Table 5, on remarque que le nombre de clusters différentiels, après correction de la p-valeur, baisse pour tous les chromosomes, et ce, pour chaque comparaison. On remarque aussi que pour chaque chromosome le nombre de clusters différentiels reste proche du nombre moyen, pour ES vs CN entre 100% et 75.8%, pour ES vs NPC entre 88.1% et 40.9% et pour CN vs NPC entre 38.5% et 8.7%. En réalisant cette comparaison, on observe que le chromosome avec le nombre de clusters différentiels est le chromosome 12.

Le chromosome 12 avait déjà marqué notre attention car il s'agit, aussi, du chromosome avec le moins de clusters et maintenant, il s'agit aussi du chromosome avec le plus de clusters différentiels. Le fait que la chromosome 12 est celui qui a le moins de clusters et qu'il s'agit du douzième chromosome le plus grand, peut indiquer que les clusters de ce chromosome sont plus grands que ceux des autres chromosomes. Une hypothèse peut être alors formulée, suggérant que la taille des arbres influent sur la différentialité des clusters. Donc, plus les arbres d'un cluster sont grands, plus ils auraient de chances d'être différentiels. Pour vérifier cela, nous nous intéressons au nombre moyen de feuilles des arbres par chromosomes, qui est présenté dans la Table 6.

chromosome	ES vs CN		ES vs NPC		CN vs NPC	
	clusters différentiels après BH (en %)	nombre moyen de feuilles	clusters différentiels après BH (en %)	nombre moyen de feuilles	clusters différentiels après BH (en %)	nombre moyen de feuilles
1	81.7	23.4	59.4	24.0	29.7	26.0
2	87.3	22.7	58.7	23.0	30.2	25.6
3	84.0	25.1	56.6	24.3	22.0	28.7
4	87.8	31.2	65.3	30.3	33.0	33.6
5	84.7	26.7	68.4	30.2	25.0	28.5
6	91.5	27.6	55.2	27.9	27.6	29.9
7	85.0	28.3	60.0	28.3	36.6	28.0
8	83.7	29.3	73.3	29.3	35.4	31.9
9	82.8	20.9	54.9	19.9	25.5	23.8
10	81.1	26.8	69.4	26.0	19.1	27.1
11	91.3	22.9	63.8	22.6	15.0	22.2
12	100.0	39.6	88.1	39.6	38.5	45.0
13	91.1	29.7	65.0	29.3	21.8	30.1
14	83.3	31.1	65.8	31.9	20.3	30.7
15	75.8	21.3	40.9	21.7	14.0	23.5
16	83.0	21.6	50.0	20.6	18.5	23.4
17	80.6	25.6	70.1	27.5	32.9	23.3
18	79.7	22.2	54.8	20.9	8.7	25.4
19	76.9	17.9	56.7	17.4	11.5	19.1

Table 6 : Table du pourcentage de cluster différentiels après correction de la p-valeur pour chaque chromosome et le nombre moyen de feuilles des sous-arbres par cluster.

Sur la Table 6, on observe bien que les arbres des clusters du chromosome 12 sont plus grands avec entre 45 à 39.6 feuilles en moyenne. En observant, les autres résultats, il semble y avoir une corrélation entre le nombre moyen de feuilles et le nombre de clusters différentiels. Pour une meilleure visualisation des résultats, ils sont représentés sous forme de nuage de points dans la Figure 10, puis la corrélation entre le nombre moyen de feuilles et le nombre de clusters différentiels.

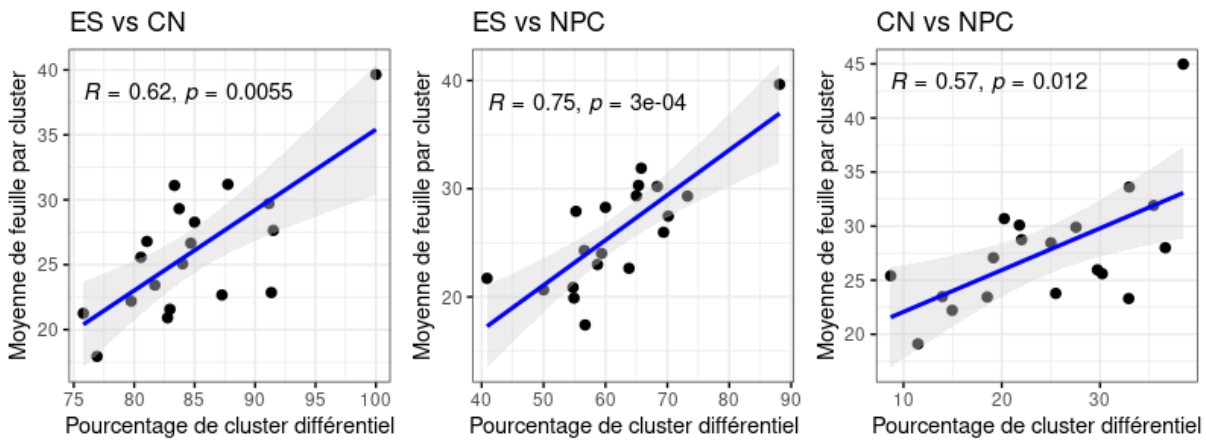

Figure 10: Graphique du nombre du pourcentage de cluster différentiel par chromosome et du nombre moyen de feuilles par cluster des trois conditions de type cellulaire.

La Figure 10, des trois nuages de point montre qu'il y a une corrélation entre le nombre de clusters différentiel et le nombre moyen de feuilles par clusters. Cependant, le chromosome 12 (point en haut à droite de chaque graphique) semble particulier.

Pour appuyer les propos des tests de corrélation entre la taille moyenne des arbres par chromosomes et le pourcentage de clusters différentiels sont réalisés, un test par comparaison est réalisé. Le test de corrélation de Spearman est réalisé pour les trois comparaisons. Le résultat des tests est que l'hypothèse nulle est refusée, la corrélation est différente de 0. Les coefficients de corrélation sont, dans l'ordre des comparaisons, de 0.62, 0.75 et 0.57.

Les tests ont aussi été réalisés en enlevant les résultats du chromosome 12. L'influence de ce chromosome sur la corrélation semble forte. Les tests de Spearman ont été effectués. Pour les trois tests, l'hypothèse nulle est toujours refusée mais les coefficients de corrélation varient, 0.55, 0.70, 0.49. Pour ces tests, l'estimation de la corrélation est plus faible.

Discussion

L'analyse des données de souris

La première étape, l'importation des données, a permis de convertir les fichiers du format Hi-C - Pro en objet HiCDOCDataSet. La deuxième étape est la normalisation des matrices de comptage.

Normalisation

Une normalisation est réalisée sur les matrices de comptages obtenues. La normalisation à pour but d'enlever les possibles biais. Pour ce type de données, un biais lié à la profondeur des paires de lectures peut exister. En effet, avant le séquençage, les lectures sont amplifiées afin d'obtenir une bonne couverture pour l'alignement du génome. Lors de cette amplification, certaines lectures sont plus amplifiées que d'autres et lors du comptage, ces lectures sont donc plus nombreuses que les autres. L'étape de normalisation permet d'éviter ce phénomène.

Avant de normaliser, pour les matrices provenant de cellules nerveuses de souris, on remarque que les matrices avec le plus de comptages de paires de bins ont une valeur moyenne de comptage plus grande que les autres et inversement. Il y a donc un biais de profondeur. Cependant, ce biais semble faible.

Après la normalisation, les matrices de comptage ont une valeur moyenne de comptage plus faible. La normalisation permet de baisser le comptage des paires de bins. En baissant, les paires de bins qui ont été séquencé en plus grande quantité dû au biais de profondeur sont réduit. En effet, on remarque que les valeurs maximales de paire de bins diminuent dans les trois comparaisons. Donc la normalisation semble bien fonctionner.

Cette étape a permis d'observer que le nombre moyen de comptage par paire de bins est influencé par le nombre de paires de bins totale. La normalisation a permis de baisser le nombre de comptage par paire de bins.

Obtention des arbres et clusters

Une fois normalisées les matrices de comptage de chaque chromosome sont transformées en arbres puis coupées en sous-arbres pour former des clusters. Le nombre de clusters et de sous-arbres obtenus est ensuite recensé dans la table 2.

Les comparaisons de cellules ES/CN et ES/NPC ont le même nombre de clusters, 1877, et donc de sous-arbres, 15016. Ces résultats pourraient indiquer que le résultat de ces deux comparaisons est assez proche. En effet, si le nombre de clusters est identique c'est que le découpage des arbres est similaire, ce qui pourrait induire que la distribution des comptages est semblable. De plus, les deux comparaisons contiennent les cellules ES, la distribution des comptages des cellules ES est la même pour les deux comparaisons, mais cela impliquerait que les distributions des comptages de CN et NPC seraient aussi sûrement similaires. Cette information du nombre de clusters permet une première supposition sur la proximité des structures génomiques des cellules CN et NPC. Le nombre de clusters de la comparaison CN et NPC est de 1757 et avec 14056 sous-arbres. Il y a moins de clusters pour cette comparaison, cela peut peut-être indiquer que l'arbre consensus de chaque chromosome est plus homogène. On peut donc supposer que c'est dû à la proximité des deux types cellulaires.

En regardant le nombre de clusters et de sous-arbres de chaque chromosome, on remarque que le nombre de clusters entre les comparison ES/CN et ES/NPC n'est pas le même pour les tous les chromosomes, donc, le découpage des arbres des chromosomes n'est pas le même pour les deux comparaisons. Les nombres de clusters sont quand même proches. On peut supposer que les clusters sont quand même similaires. Les comparaisons ES/CN et ES/NPC sont proches mais peut-être pas aussi que le laissait présager le résultat général.

Avec ces résultats, on remarque aussi que plus un chromosome est grand plus il y a de clusters, sauf pour le chromosome 12, qui est celui avec le moins de clusters. Les arbres des chromosomes sont moins divisés. Ce résultat laisse présager que ce chromosome 12 est peut-être particulier et que sa structure est différente des autres.

Les résultats de cette étape sont similaires entre les comparaisons intégrant le type cellulaire ES. Le nombre de clusters diminue avec la taille du chromosome sauf pour le chromosome 12.

Analyse différentielle

La dernière étape réalisée avec le package est l'analyse différentielle, une p-valeur pour chaque cluster de chaque comparaison cellulaire est obtenue. Ces p-valeurs sont ensuite corrigées avec le processus de Benjamini-Hochberg.

Chaque cluster de chaque comparaison a été comparé. Le résultat principal est le nombre de clusters différentiels pour chaque comparaison. La comparaison ES/CN est celle avec le pourcentage le plus élevé de 85%, la deuxième comparaison la plus élevée est ES/NPC avec 61.2% et la dernière avec 24.1% est CN/NPC. Un cluster différentiel signifie que les sous-arbres des deux comparaisons sont significativement différents. Un sous-arbre représente la conformation spatiale d'une partie d'un chromosome. Donc si un cluster est différentiel cela signifie que la conformation spatiale des types cellulaires est différente pour la partie du chromosome que prend en compte le cluster. Les résultats montrent que la comparaison ES/CN est celle avec le plus de clusters différentiels. Donc cette comparaison est celle avec le plus de parties du génome différentes entre les deux types cellulaires. On peut donc dire que les types cellulaires les plus éloignés sont ES et CN. Avec le même raisonnement, on peut dire que les cellules CN et NPC ont la conformation génomique la plus proche.

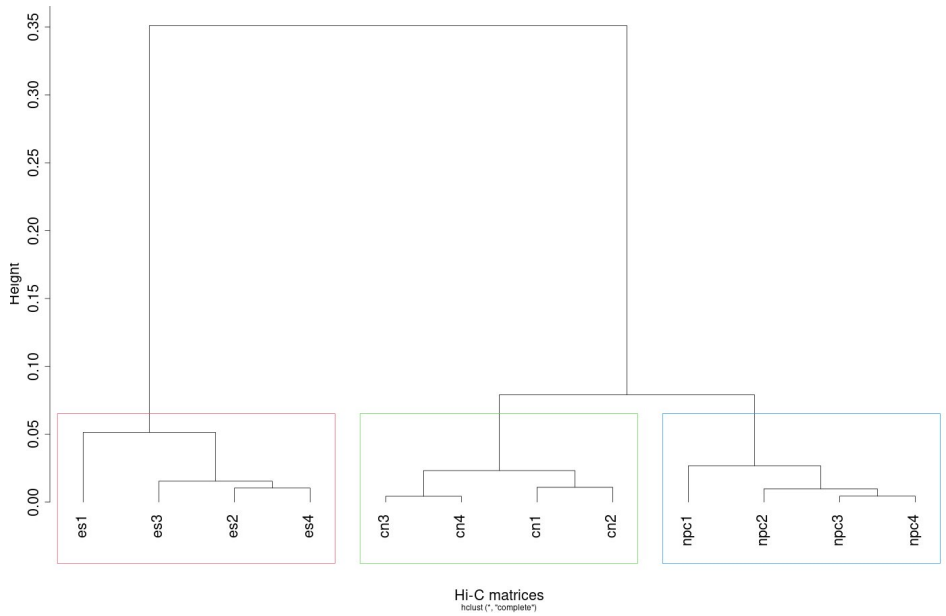

Figure 11 : Clustering hiérarchique des matrices des 19 chromosomes de chaque réplicats biologiques des trois type cellulaire

Ces résultats semblent corroborer avec une analyse précédemment réalisée sur les mêmes données, qui a produit un arbre de clustering hiérarchique (Figure 11). Cet arbre a été obtenu avec le package HiCRep, qui permet de calculer une corrélation pondérée entre matrices Hi-C et donc d'estimer leur similarité. Cet arbre représente donc la proximité des matrices des différents réplicats. On constate que les réplicats entre types cellulaires sont les plus proches, comme attendus. On remarque que les réplicats des cellules CN et NPC sont les plus proches, car la distance des branches est plus petite entre eux, que celles du type cellulaire ES. La dernière information confirmée est que le type cellulaire ES est plus différent des deux autres. Cet arbre ne permet pas de déterminer quel type cellulaire est plus proche de ES, en effet la taille des branches entre ES/CN et ES/NPC est égale.

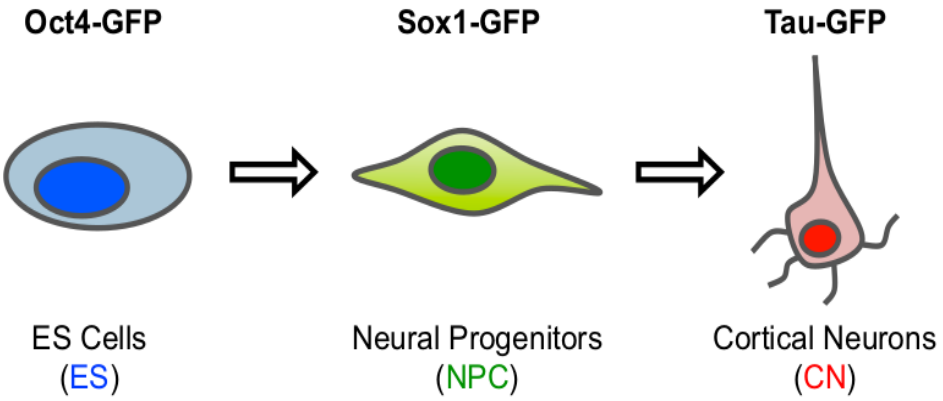

Figure 12 : Schéma du processus de différenciation des cellules souches embryonnaires, des cellules souche neurales et des cellules de neurones corticaux

L'analyse des données de souris est, aussi, cohérente avec ce qui a été dit dans la littérature scientifique (Figure 12), et notamment dans la publication de Bonev, 2017[13]. Cette publication permet de confirmer que NPC et CN sont les types cellulaires avec la conformation tridimensionnelle la plus proche mais aussi que les cellules ES et CN sont les types avec la conformation la plus différente.

Les cellules souches se différencient en NPC puis en CN il est donc normal de retrouver que les cellules souche (ES) soient plus proches des NPC que des cellules CN. De plus, les cellules CN et NPC sont des cellules différenciées contrairement aux cellules (ES), elles sont, donc, plus proches entre que des cellules ES.

Les résultats obtenus grâce à la méthode treediff sont cohérents avec ceux présents dans la littérature scientifique, soulignant la pertinence de la méthode.

Le dernier résultat qui peut être discuté est la corrélation entre le nombre de feuilles par clusters et la p-valeur obtenue. Il semble qu'il y ait un biais qui est observé entre la taille des arbres et le nombre de clusters différentiels. En effet, plus les arbres sont grands, plus ils sont différentiels, une tendance confirmée par un test de corrélation de Spearman. Cette information est à prendre en compte pour la méthode treediff car cela pourrait fausser les résultats. En effet, cela voudrait dire que le résultat final dépend de la taille des clusters et pas seulement de la comparaison des arbres.

Pour rectifier ce biais, deux axes de recherche peuvent être étudiés. Le premier est celui de la méthode pour découpage des arbres en sous-arbres. La méthode actuelle est le "broken stick" qui utilise la distribution des distances de paires de feuilles. Une méthode plus précise en fonction de la topologie des arbres pourrait être utilisée. Mais même avec une autre méthode cela n'enlèvera pas, l'influence de la taille de l'arbre sur la différentialité.

La seconde option est de s'intéresser à la méthode d'agrégation des p-valeurs, Simes. Cette méthode prend la plus petite valeur obtenue pour déterminer la différentialité du cluster. Bien que cette méthode d'agrégation prenne soin de multiplier les p-valeurs par le nombre de tests, dans le but de prendre en compte la quantité d'information, l'ajustement réalisé ne semble pas parfaitement adapté aux données utilisées ici. La méthode d'agrégation semble donc la première partie de la méthode à améliorer.

Pour terminer, bien qu'il y ait un biais avéré dû à la taille des clusters, on remarque que les résultats biologiques sont en accord avec les autres études sur

ces types cellulaires. Le biais n'empêche pas d'avoir de bons résultats. En effet, ce biais étant le même sur les trois comparaisons testées, il affecte les résultats de la même manière. Cette méthode peut donc être utilisée pour réaliser une analyse différentielle de données Hi-C.

Conclusion/perspectives

La package treediff est disponible sur le CRAN, c'est un package qui contient cinq fonctions qui permettent de réaliser une analyse de données Hi-C entre deux conditions biologiques. Le package est fonctionnel et permet de réaliser des analyses dans un temps correct. L'utilisation est accessible grâce à une documentation.

Le package a permis de réaliser l'analyse de données Hi-C issue de souris, correspondant à trois types de cellules nerveuses ES, NPC et CN. La comparaison avec le plus de clusters différentiels est ES/NPC et celle avec le moins est CN/NPC. Une corrélation est trouvée entre le nombre moyen de feuilles par clusters et le nombre de clusters différentiels.

En observant les matrices avant normalisation, un biais de profondeur de séquençage est observé. Après la normalisation, le biais semble corrigé. Les résultats de l'étape de l'obtention des clusters et des sous-arbres permet de faire une première conjecture sur la proximité des types cellulaires NPC et CN et, donc, sur leur différence avec les cellules ES. Cette conjecture est confirmée avec le résultat de l'analyse différentielle. Les cellules CN et NPC ont une conformation tridimensionnelle la proche, parmi les trois types cellulaires. L'analyse différentielle permet aussi de conclure que les cellules CN et ES ont l'information tridimensionnelle la plus éloignée. Ces conclusions sont cohérentes avec ce qui est connu par ailleurs concernant les lignées cellulaires étudiées.

La dernière information à retirer de cette analyse est le biais trouvé entre le nombre moyen de feuilles par cluster et le nombre de clusters différentiels. Pour gommer ce biais, la méthode d'obtention des clusters peut être revue mais cela n'empêche pas le biais de perdurer. Il semble que la méthode d'agrégation des p-valeurs est perfectible, ce qui donne une piste à explorer pour la suite des travaux de recherche.

La package treediff permet de faire l'analyse des données de l'importation des données jusqu'à l'obtention des p-valeurs de l'analyse différentielle. Une fois les p-valeurs obtenues, il faut les corriger puis réaliser une analyse de celle-ci. Cette dernière partie de l'analyse peut être une suite afin de finaliser l'implémentation de ce package. De plus, d'autres méthodes d'obtention des clusters peuvent être ajoutées ou encore une autre manière d'agréger les p-valeurs.

Pour terminer, le chromosome 12 a permis de mettre en évidence un biais de la méthode. Mais ce biais semble bien plus fort pour ce chromosome. Il pourrait donc être intéressant de réaliser une analyse plus approfondie de ce chromosome en particulier afin de mieux comprendre sa spécificité.

Références

1. Vogelstein, B. Cancer Genome Landscapes.
2. Li, G. *et al.* Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. *Cell* **148**, 84–98 (2012).
3. Dixon, J. R. *et al.* Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**, 376–380 (2012).
4. Bonev, B. & Cavalli, G. Organization and function of the 3D genome. *Nat. Rev. Genet.* **17**, 661–678 (2016).
5. Kempfer, R. & Pombo, A. Methods for mapping 3D chromosome architecture. *Nat. Rev. Genet.* **21**, 207–226 (2020).
6. Lieberman-Aiden, E. *et al.* Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science* **326**, 289–293 (2009).
7. Rao, S. S. P. *et al.* A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell* **159**, 1665–1680 (2014).
8. Stansfield, J. C., Cresswell, K. G., Vladimirov, V. I. & Dozmorov, M. G. HiCcompare: an R-package for joint normalization and comparison of HI-C datasets. *BMC Bioinformatics* **19**, 279 (2018).
9. Lun, A. T. L. & Smyth, G. K. diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics* **16**, 258 (2015).
10. Weinreb, C. & Raphael, B. J. Identification of hierarchical chromatin domains. *Bioinformatics* **32**, 1601–1609 (2016).
11. Ambroise, C., Dehman, A., Neuvial, P., Rigaill, G. & Vialaneix, N. Adjacency-constrained hierarchical clustering of a band similarity matrix with application to genomics. *Algorithms Mol. Biol.* **14**, 22 (2019).
12. Marti-Marimon, M. *et al.* Major Reorganization of Chromosome Conformation During Muscle Development in Pig. *Front. Genet.* **12**, (2021).
13. Bonev, B. *et al.* Multiscale 3D Genome Rewiring during Mouse Neural Development. *Cell* **171**, 557-572.e24 (2017).
14. Bennett, K. D. Determination of the number of zones in a biostratigraphical sequence. *New Phytol.* **132**, 155–170 (1996).
15. Smyth, G. K. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Stat. Appl. Genet. Mol. Biol.* **3**, (2004).
16. Simes, R. J. An improved Bonferroni procedure for multiple tests of significance. *Biometrika* **73**, 751–754 (1986).
17. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B Methodol.* **57**, 289–300 (1995).

18. Wickham, H. testthat: Get Started with Testing. *R J.* **3**, 5 (2011).
19. Wickham, H., Danenberg, P., Csárdi, G. & Eugster, M. roxygen2: In-Line Documentation for R. https://roxygen2.r-lib.org/ (2022).
20. Wickham, H., Hester, J., Chang, W. & Bryan, J. devtools: Tools to Make Developing R Packages Easier. https://devtools.r-lib.org/ (2022).

Gwendaëlle Cardenas Année universitaire : 2022 - 2023

Master de Bioinformatique de l'Université de Rennes 1

Pour entrer dans une cellule, le génome se condense. Cette condensation n'est pas aléatoire, elle suit une structure hiérarchique. En étant condensées, des parties du génome éloignées sur la séquence se retrouvent suffisamment proches pour interagir entre elles. Ces intéractions peuvent influencer l'expression génique, donc un changement de conformation tridimensionnelle du génome peut avoir une influence sur le fonctionnement de l'organisme.

Une méthode pour étudier cette conformation est la technique Hi-C. Les intéractions spatiales du génome sont agrégées dans une matrice de comptage. Pour comparer, les différences de conformation de deux conditions biologiques, une méthode d'analyse différentielle est utilisée, treediff. La méthode utilisée, durant le stage, permet de comparer les différences entre deux ensembles de matrices, en conservant la structure hiérarchique du génome.

Le premier objectif du stage est l'implémentation de cette méthode dans un package R, treediff, à partir des scriptes qui ont permis de tester la méthode. Le package est disponible sur le CRAN. Le second objectif est la réalisation de l'analyse différentielle de données de souris. Les données utilisées sont celles de trois types cellulaires : les cellules souches embryonnaires (ES), les cellules souches neurales (NPC) et les cellules de neurones corticaux (CN). L'analyse différentielle indique que les cellules CN et NPC ont la conformation tridimensionnelle la plus proche et ES et CN la plus éloignée. En effet, il s'agit de trois types de cellules appartenant à un processus de différentiation. Les cellules ES se différencient en cellules NPC qui se différencient en cellules CN.

Mot-clés : Hi-C, analyse différentielle, treediff, R, arbre de clustering hiérarchique

To enter into a cell, the genome condenses. This condensation is not random, but follows a hierarchical structure. By being condensed, parts of the genome that are far apart in sequence find themselves close enough to interact with each other. These interactions can influence gene expression, so a change in the genome's three-dimensional conformation can influence how the organism functions.

One method of studying this conformation is the Hi-C technique. The genome's spatial interactions are aggregated in a counting matrix. To compare conformational differences between two biological conditions, a differential analysis method is used. For this internship, a new method is used to compare the differences between two sets of matrices, while preserving the hierarchical structure of the genome.

The first objective of the internship is to implement this method in an R package, treediff, based on the scripts used to test the method. The package is available from CRAN. The second objective is to carry out a differential analysis of mouse data. The data used are those from three cell types: embryonic stem cells (ES), neural progenitors (NPC) and cortical neurons (CN). Differential analysis indicates that CN and NPC cells have the closest three-dimensional conformation, and ES and CN the furthest apart. In fact, these are three cell types involved in a differentiation process. ES cells differentiate into NPC cells, which in turn differentiate into CN cells.

Keywords: Hi-C, differential analysis, treediff, R, hierarchical clustering tree