Rapport Projet M1 BIM
Etude de la diversité chez une population de lymphocytes B dans un contexte clinique

DRADJAT Kevin
FALL Assane
Encadré par J.Silva-Bernardes

Sorbonne Université 2022

Sommaire

Chapitre 1

Introduction

Le séquençage de nouvelle génération (Next Generation Sequencing : NGS) a donné à la communauté scientifique un accès à de nouvelles possibilités en termes d'analyses de séquences. En effet, la NGS peut produire un nombre important de données sous forme de séquence, dont des séquences de récepteurs de lymphocytes B qui représentent le répertoire immunitaire d'un individu.

On peut analyser ces données en étudiant la diversité au sein de ce répertoire. L'étude de la diversité au sein d'un répertoire immunitaire permet plusieurs choses :

— Identifier et classer des répertoires en fonction de leurs caractéristiques. On pourra donc identifier un individu sain ou malade.

— Etudier les effets d'un vaccin ou d'un traitement en analysant le répertoire immunitaire d'un individu au cours du temps

Pour quantifier et mesurer cette diversité, nous utilisons des outils mathématiques et informatiques. Malgré les nombreux outils disponibles permettant d'effectuer cette tâche, la complexité et la nécéssité de combiner plusieurs outils rendent l'étude de la diversité compliquée. Cette méthode n'est donc pas souvent utilisée pour identifier un individu sain ou malade.

Le but de ce projet est d'étudier et d'identifier les outils informatiques et modèles mathématiques les plus intéressants afin d'utiliser l'étude de la diversité en contexte clinique.
Nous avons aussi pour objectif d'implémenter ces méthodes et de les intégrer à une plateforme qui facilite leur utilisation.

Chapitre 2

Outils en R

Notre objectif est de développer des outils en Python permettant d'analyser la diversité d'un répertoire immunitaire et de comparer deux répertoires.

Nous nous sommes basés sur des packages R qui sont utilisés pour effectuer ces tâches : ImmuneRef[5], SumRep[4], DivE[3], ReCoLD[2] et RDI[1]. Nous avons sélectionné les packages utiles et nous avons alors converti les fonctions les plus intéressantes de ces packages en Python. Nous avons également développé de nouveaux outils en nous inspirant de ceux-ci. Une description des packages est présentée dans ce chapitre.

2.1 ImmuneREF

ImmuneRef[5] est un outil sous forme de package R permettant d'analyser un répertoire immunitaire en donnant 6 mesures de diversité interprétables :

- L'indice de diversité de Hill.
- Les fréquences d'acides aminés par positions sont calculées dans chaque séquence
- Le nombres de k-mers, contenant m gaps ou moins de m gaps, dans chaque séquence.
- Répartition des gènes IGHV, IGHD et IGHJ.
- Pourcentage de similarité entre deux répertoires.
- Etablissement d'un réseau où chaque nœud représente une séquence de récepteur, connectés par des arêtes calculés avec la distance de Levenshtein.

2.2 SumREP

SumRep[4] est un package R permettant de calculer plusieurs valeurs statistiques concernant un répertoire immunitaire. Cet outil prend en entrée une base de données de séquences de récepteurs de lymphocytes B et T sous format AIRR.
SumRep peut servir à différentes choses, comme détecter des motifs dans un alignement de séquences, ou encore des réarrangements entre séquences. SumRep est également utile pour comparer deux répertoires. SumRep dispose d'outils graphiques permettant de visualiser ces valeurs au sein d'un répertoire ou entre deux répertoires.

2.3 DivE

L'outil Diversity Estimator[3] (DivE) est une méthode heuristique pour estimer le nombre d'espèces en fonction du nombre d'individus. En effet, un échantillon d'une population ne représente pas toute la diversité de la population. La résolution de ce problème d'espèces non-observées est importante dans le domaine de l'immunologie. L'application de cette méthode à une population de lymphocytes permet de mieux comprendre les réponses immunitaires et permet d'estimer la diversité de cette population. L'estimateur DivE s'appuie sur plusieurs modèles mathématiques, fournis avec le package R, pour prédire la croissance du nombre d'espèces. Ces modèles sont appliqués à des échantillons de la population donnée et sont classés selon plusieurs critères :

— Divergence : Erreur moyenne entre les données et les prédictions. Le modèle doit prédire les données sur lesquels on l'a appliqué.

— Précision : À partir d'un sous-échantillon, le modèle doit prédire correctement l'évolution du reste de l'échantillon.

— Similarité : Le modèle doit effectuer des prédictions similaires à partir de n'importe quel sous-échantillon.

— Plausibilité : Le nombre d'espèces ne doit pas décroître et le taux d'apparition d'espèces ne doit pas augmenter. Autrement dit, on doit avoir , $\forall$ x $\geq$ 1, S'(x) $\geq$ 0 $\wedge$ S"(x) $\leq$ 0, où S est la fonction qui associe le nombre d'espèces au nombre d'individus.

Grâce à ces critères, on associe un score à chacun des modèles. On sélectionne les cinq meilleurs modèles et on effectue la moyenne géométrique de leur prédiction (nombre d'espéces sur la population totale) pour donner une bonne estimation.

Chapitre 3

Développement des Outils en Python

Notre objectif est de sélectionner les fonctionnalités les plus intéressantes de ces packages R pour les implémenter en Python. On distingue deux types de fonctionnalités : celles qui calculent une caractéristique pour un répertoire et celles qui comparent deux ou plusieurs répertoires à travers une caractéristique.
Ces caractéristiques doivent permettre de décrire explicitement la diversité d'un répertoire et ainsi permettre de différencier plusieurs types de répertoires.

Les tests ont été effectués sur deux types de répertoires simulés :

- Monoclonal : répertoire d'une personne malade
- Polyclonal : répertoire d'une personne saine

3.1 Caractétistiques d'un Répertoire

3.1.1 Indice de Diversité de Hill

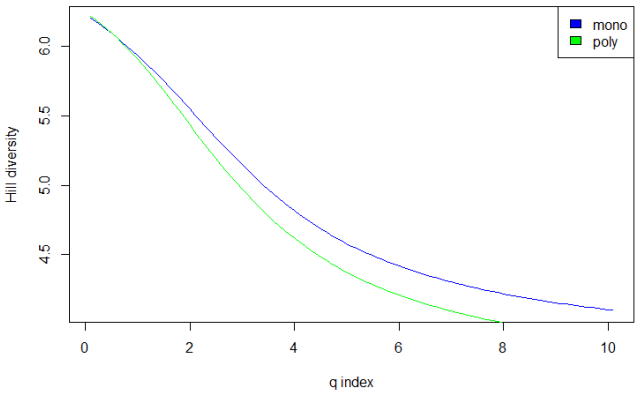
L'indice de diversité de Hill pour une population de N espèces, où p_i est la proportion de l'espèce i, se calcule avec la formule suivante :

$$ {}^{q}H = \left(\sum_{i=1}^{N} p_i^q \right)^{\frac{1}{1-q}} $$

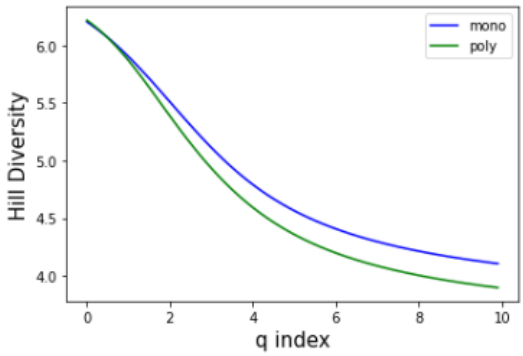
On se base sur le package SumRep. On considère comme espèces les séquences avec leurs proportions respectives dans le répertoire. On calcule cette valeur pour q compris entre 0 et 10 avec un pas de 0,1.
Pour obtenir les mêmes résultats avec l'implémentation en Python, nous calculons le logarithme de l'indice de diversité.

Nous traçons les courbes d'évolution de l'indice de diversité en fonction de q. Nous pouvons alors analyser ces courbes pour déterminer le type des répertoires. Plus la courbe tend rapidement vers 0 et moins il y a de diversité dans le répertoire associé.

Sur la Figure 3.1 nous voyons que la courbe du répertoire monoclonal est au-dessus de la courbe associée au répertoire polyclonal, ce qui impliquerait que le répertoire monoclonal est moins diversifié que le répertoire polyclonal. Or, le répertoire monoclonal admet moins de diversité car il est associé à un individu malade. L'outil donne donc des incohérences dans ce cas.
Cette incohérence est dûe au fait que l'on considère comme espèces les séquences se répétant. Nous considérerons par la suite comme espèces les clones pour obtenir des résultats cohérents. Les données simulées fournies ne possédant pas cette annotation, nous ne pouvons pas effectuer les tests sur celles-ci.



ImmuneREF



Python

FIGURE 3.1 – Comparaison des résultats pour le calcul de l'indice de diversité de Hill en fonction de α entre ImmuneREF et l'implémentation en Python.

3.1.2 Pairwise Distance Distribution (PDD)

Le calcul de la Pairwise Distance Distribution effectué avec le package SumRep utilise la distance de Levenshtein, qui correspond au nombre minimal de caractères à supprimer, insérer ou remplacer pour passer d'une chaîne de caractères à une autre.
Cette distance se calcule avec la formule de récurrence suivante :

$$l(a,b) = \begin{cases} \max(|a|,|b|) & \text{si } \min(|a|,|b|) = 0, \\ \text{lev}(a-1,b-1) & \text{si } a[0] = b[0], \\ 1 + \min \begin{cases} \text{lev}(a-1,b) \\ \text{lev}(a,b-1) \\ \text{lev}(a-1,b-1) \end{cases} & \text{sinon.} \end{cases}$$

Pour l'implémentation en Python, nous avons donc créé une fonction permettant de calculer la distance de Levenshtein entre deux chaînes de caractères.
Il suffit ensuite de calculer cette distance pour chaque paire de séquence.
Cependant, l'utilisation de la fonction implémentée pour le calcul de toutes les distances implique un temps d'exécution trop grand. Nous avons donc utilisé le module Levenshtein en Python pour réduire le temps d'exécution.

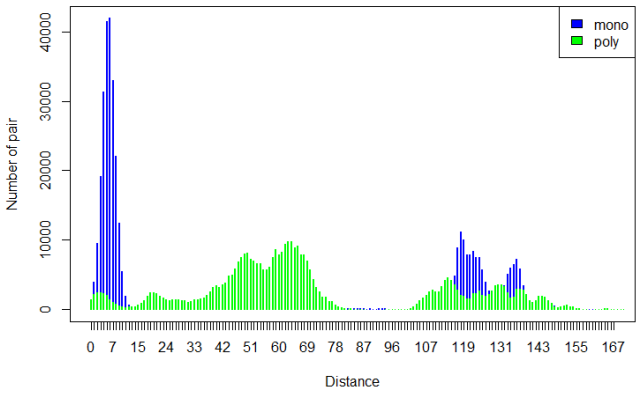
Sur la Figure 3.2, nous remarquons que le répertoire monoclonal possède beaucoup de séquences similaires grâce au pic proche de 0 visible sur son histogramme. Nous remarquons également que le répertoire polyclonal est plus diverfié que le répertoire monoclonal, avec des distances centrées à environ 60 et 120 et un pic moins élevé en 0.

Approximation et Amélioration

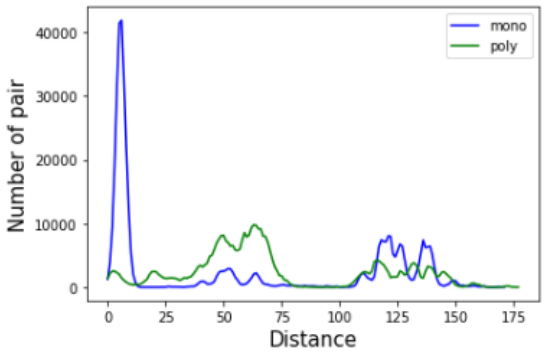
Nous avons développé plusieurs versions de la fonction permettant de calculer la PDD.

La première version de la fonction calculant la PDD consiste à considérer toutes les séquences du répertoire et calculer la distance de Levenshtein pour chaque paire de séquences. Le problème est que si le répertoire contient beaucoup de séquences le temps de calcul sera conséquent. En effet, si on considère n séquences on doit alors calculer $\frac{n(n-1)}{2}$ valeurs de distances.
Nous avons donc fait le choix d'échantillonner le répertoire considéré à 10000 séquences pour atteindre un temps d'exécution tournant autour des 10 minutes pour un répertoire. Notons que le package SumRep fait également ce choix lorsque le répertoire fait plus de 10000 séquences.

Nous avons également fait le choix de mettre la proportion de paires de séquences en ordonnées au lieu du nombre de paires de séquences pour ignorer la taille du répertoire considéré.



SumRep



Python

FIGURE 3.2 – Comparaison du nombre de paires de séquences en fonction de la distance de Levenshtein, entre SumRep et l'implémentation en Python.

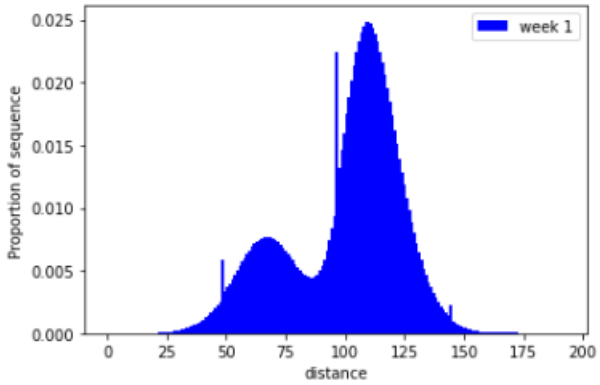


FIGURE 3.3 – Pairwise Distance Distribution sur un échantillon de 10000 séquences.

Nous avons codé une deuxième version de la fonction permettant de calculer la PDD dans le but de réduire le temps d'exécution. On remarque que la première version de la fonction donne des résultats précis, même en échantillonnant à 10000 séquences. Or, on cherche seulement à identifier la tendance de la courbe de répartition. On peut donc penser à approximer encore plus les résultats pour diminuer le temps d'exécution.

Nous avons fait le choix de partitionner le répertoire considéré en échantillons de 1000 séquences et calculer les PDD de chacun de ces échantillons. (Figure 3.4)
Lors de cette approximation, les distances entre les séquences qui ne sont pas dans la même partition ne seront pas calculées, ce qui rend cette méthode moins précise.

On peut approximer encore plus les résultats en effectuant un premier échantillonnage pour considérer 10000 séquences et ensuite partitionner cet échantillon en sous-échantillons de 1000 séquences et calculer la PDD de chacun de ces sous-échantillons. Cela revient à échantilloner 1000 séquences dans le répertoire 10 fois. (Figure 3.5)

On remarque que les courbes de distributions sont similaires pour les trois approximations.

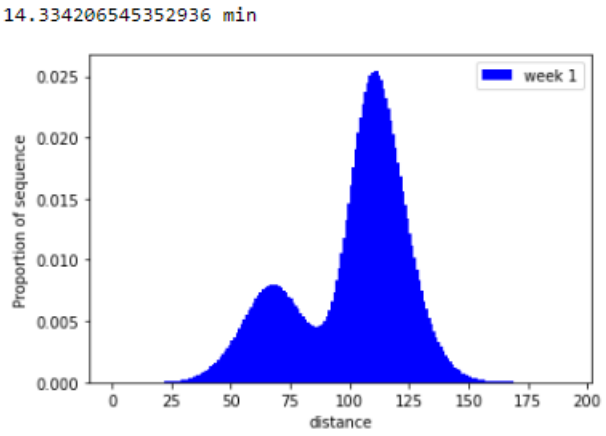


FIGURE 3.4 – Temps d'éxécution et Pairwise Distance Distribution en partitionnant le répertoire en échantillons de 1000 séquences.

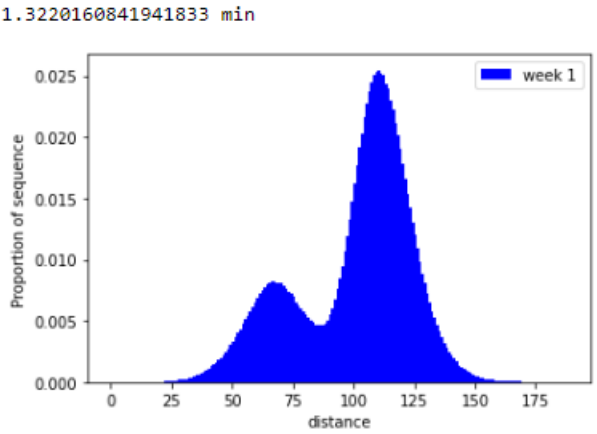


FIGURE 3.5 – Temps d'exécution et Pairwise Distance Distribution sur un échantillon de 10000 séquences partitionné en sous-échantillons de 1000 séquences.

3.1.3 Représentation en Réseau

On peut tracer un réseau sous la forme d'un graphe. Chaque nœud représente une séquence et la taille de ce nœud est proportionnelle au nombre de séquences identiques à celle-ci. Un arc relie deux nœuds si la distance de Levenshtein entre les deux séquences est inférieure à un seuil. Nous prenons un seuil de 20 par défaut.

Cette fonctionnalité n'est pas disponible dans les packages de référence et fait partie de notre contribution. On se sert de la bibliothèque Pyvis pour visualiser ces nœuds dans une fenêtre HTML.

La mise en réseau d'un répertoire de grande taille nécéssite un temps de calcul conséquent. Nous avons donc fait le choix d'échantillonner aléatoirement 5000 séquences si le répertoire contient plus de 5000 séquences.

Temps d'affichage

Le temps d'affichage pour un graphe de 5000 séquences étant assez long, on peut calculer les caractéristiques du graphe sans l'afficher. On choisit d'afficher le nombre de nœuds et d'arêtes, le degré moyen des nœuds, le nombre de composantes connexes et la taille de ces composantes connexes. (Figure 3.7)

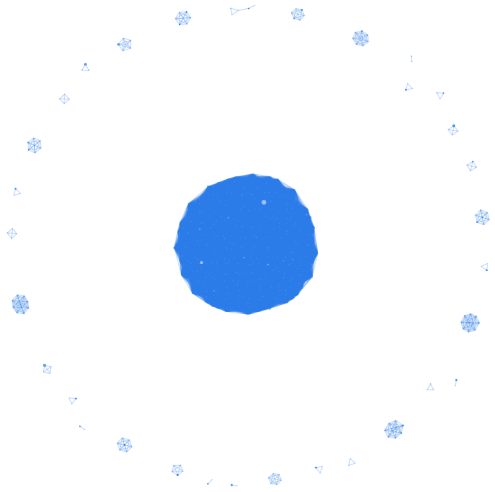


FIGURE 3.6 – Exemple de graphe représentant le répertoire monoclonal

	nodes	edges	average degree	nb of connected comp	max size	min size	average size
mono	494	59356	240.31	33	344	2	14.97
poly	501	8152	32.54	22	88	2	22.77

FIGURE 3.7 – Informations des graphes associés aux répertoires monoclonal, oligoclonal et polyclonal.

On remarque que le graphe du répertoire monoclonal possède le degré moyen le plus élevé et celui du répertoire polyclonal possède le degré moyen le moins élevé. De plus, la taille de la composante connexe la plus grande est plus élevée pour le répertoire monoclonal. Ce qui montre que les nœuds sont plus connectés entre eux dans le graphe associé au répertoire monoclonal.

3.1.4 Répartition des gènes IGHV et IGHJ

En répertoriant les gènes IGHV et IGHJ de chaque séquence dans chaque répertoire ainsi que leur proportion, on peut estimer la diversité de ceux-ci.

On peut tracer un histogramme pour visualiser les gènes V et J utilisés dans chaque répertoire. Pour la visualisation, nous ne prenons en compte que des gènes qui apparaissent plus de 100 fois dans le répertoire. On pourra changer cette valeur seuil en fonction de la taille du répertoire. Par exemple, nous avons fixé la valeur seuil à 5 car les répertoires simulés sont de petite taille.

Cette fonctionnalité n'est pas prise en compte par les packages de référence et fait donc partie de notre contribution.

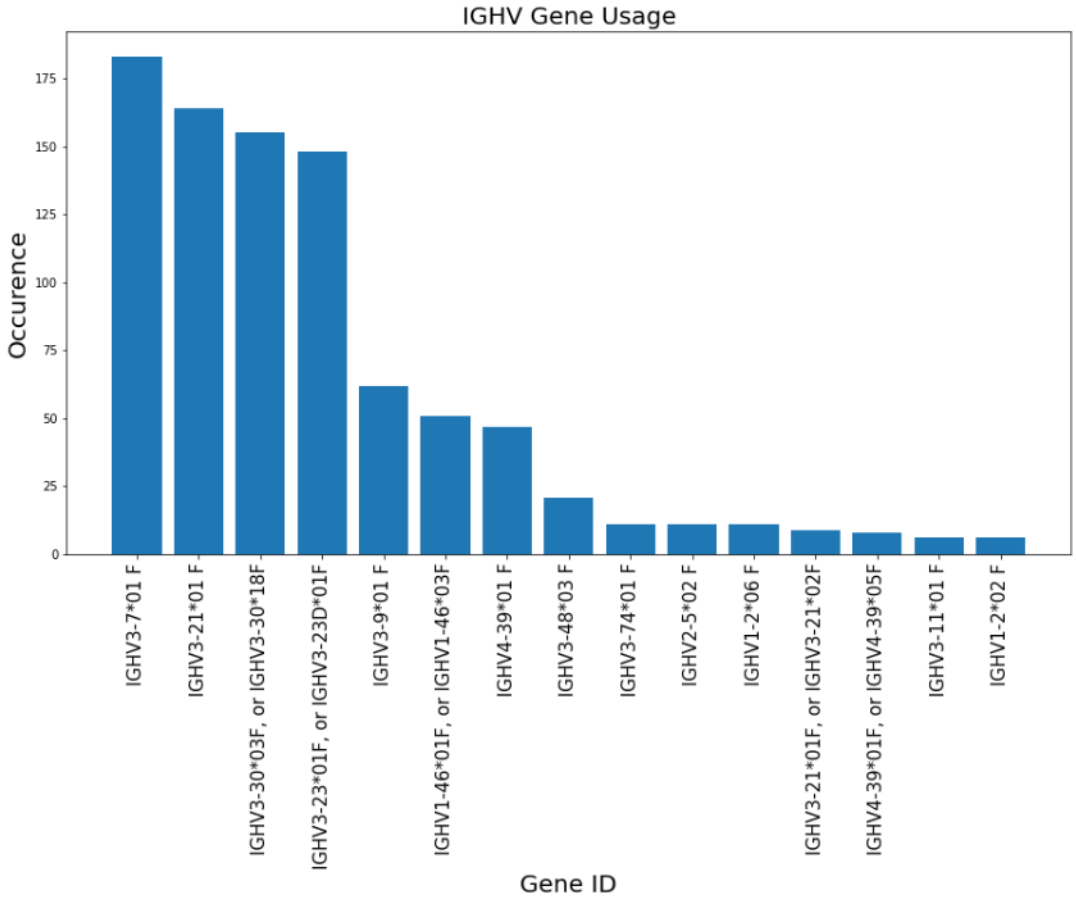


FIGURE 3.8 – Proportion des gènes IGHV dans le répertoire polyclonal avec une valeur seuil de 5.

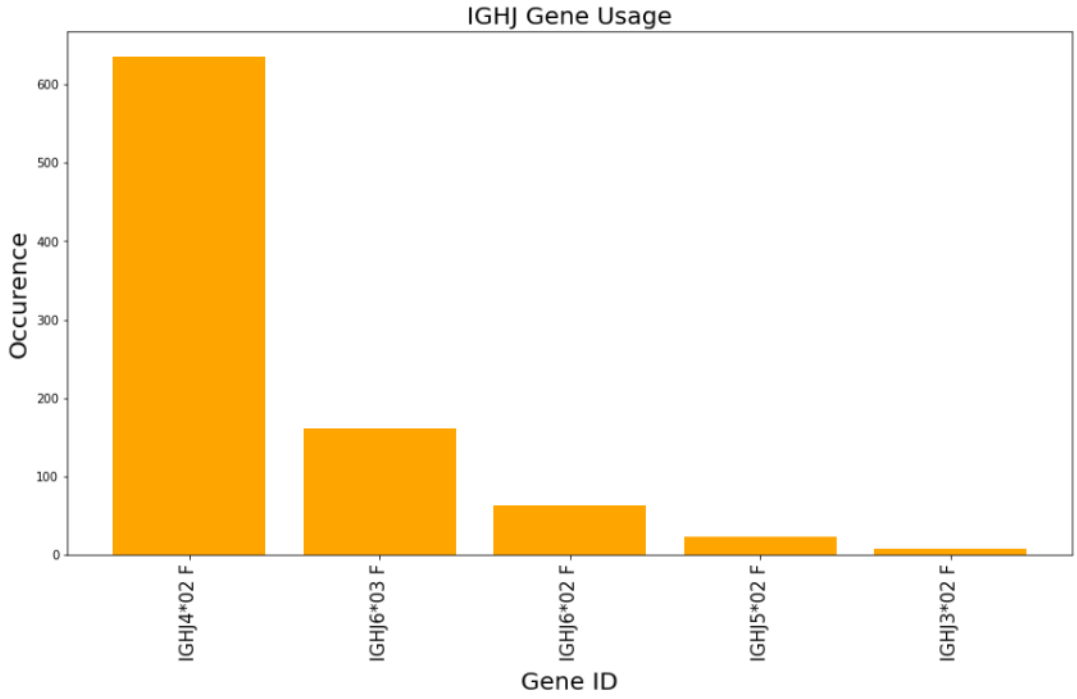


FIGURE 3.9 – Proportion des gènes IGHJ dans le répertoire polyclonal avec une valeur seuil de 5.

3.2 Comparaison de Répertoires

Nous avons identifié des caractéristiques permettant de décrire la diversité d'un répertoire. Nous allons maintenant développer des outils permettant de comparer deux ou plusieurs répertoires grâce à ces caractŕistiques.

3.2.1 Coefficient de corrélation de Pearson

La coefficient de corrélation de Pearson mesure le degré de la relation linéaire entre deux variables. Cette valeur est comprise entre -1 et 1.
Si une valeur tend à augmenter tandis que l'autre diminue, le coefficient est négatif. Inversement, si les deux variables tendent à augmenter ou diminuer toutes les deux, le coefficient est positif.
Etant donnés deux variables x et y, le coefficient de corrélation de Pearson se calcule avec la formule :

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

On calcule ce coefficient pour comparer l'évolution de l'indice de diversité de Hill de deux répertoires en fonction de la valeur de α. Plus le coefficient est proche de 1 et plus les répertoires sont similaires à l'échelle de la diversité.

On remarque que les valeurs sont très proches de 1 et mutuellement similaires. En effet, en regardant les courbes de diversité on remarque qu'elles suivent la même tendance. Cette valeur de comparaison peut donc ne pas être pertinente pour la comparaison de répertoires.

```
            mono      poly
mono 1.0000000 0.9981137
poly 0.9981137 1.0000000
```

ImmuneREF

	mono	poly
mono	1.000000	0.999387
poly	0.999387	1.000000

Python

FIGURE 3.10 – Comparaison du coefficient de corrélation de Pearson entre ImmuneREF et l'implémentation en Python.

3.2.2 Jensen-Shannon Divergence

La divergence de Jensen-Shannon (JSD) permet de comparer deux vecteurs de probabilités. Plus cette valeur est proche de 0, plus les répertoires sont similaires au niveau de la diversité.
Etant donnés deux vecteurs de probabilités P et Q, la valeur JSD se calcule avec la formule suivante :

$$JSD(P\|Q) = \frac{1}{2}(D(P\|M) + D(Q\|M))$$

Avec $M = \frac{1}{2}(P + Q)$ et $D(P\|M) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{m_i}\right)$.

Grâce à ces valeurs (Figure 3.11), on remarque que les répertoires monoclonal et polyclonal sont différents avec une valeur de comparaison éloignée de 0.

```
          mono      poly
mono 0.0000000 0.2952094
poly 0.2952094 0.0000000
```

SumRep

	mono	poly
mono	0.000000	0.290021
poly	0.290021	0.000000

Python

FIGURE 3.11 – Comparaison du calcul de la Jensen-Shannon Divergence entre les deux implémentations.

3.2.3 Pourcentage de similarité

Il s'agit du pourcentage de séquences en commun entre deux répertoires. Cette valeur est utile pour comparer l'évolution d'un répertoire immunitaire d'un patient suite à un traitement ou un vaccin.
Etant donnés deux répertoires A et B, on calcule ce pourcentage avec la formule :

$$overlap(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

Dans notre cas, on observe un pourcentage de similarité de 0% car il s'agit de données simulées ne provenant pas d'un même individu.

```
      mono poly
mono     1    0
poly     0    1
```

ImmuneREF

	mono	poly
mono	1.0	0.0
poly	0.0	1.0

Python

FIGURE 3.12 – Comparaison du calcul de pourcentage de similarité entre les deux implémentations.

3.2.4 Comparaison des gènes IGHV et IGHJ

On peut comparer les répartitions de gènes IGHV et IGHJ entre deux répertoires. Dans ce cas, on considère uniquement les gènes V et J en commun puis on trace un double histogramme avec le même axe x. (Voir Etude de cas)

Notons que cette comparaison n'est utile que pour comparer des répertoires provenant d'un même individu. En effet, des répertoires d'origine différente n'utilisent pas les mêmes gènes IGHV et IGHJ, on ne trouvera donc pas ou peu de gènes communs aux deux répertoires.

Cette fonction est utile pour observer l'évolution du répertoire immunitaire d'un même individu.

Chapitre 4

Etude de Cas

Nous allons appliquer les outils développés à une étude de cas.
Notre objectif est d'étudier les effets du vaccin contre la COVID-19 sur un patient. Nous allons considérer des répertoires obtenus par prélèvement et séquençage réalisés à plusieurs intervalles de temps. On considère trois répertoires : un premier obtenu grâce à un prélèvement juste après la vaccination, un deuxième avec un prélèvement une semaine après la vaccination et un troisième répertoire obtenu avec un prélèvement deux semaines après la vaccination.

Nous cherchons à observer une évolution de la diversité du répertoire immunitaire de l'individu.

4.1 Diversité de Hill

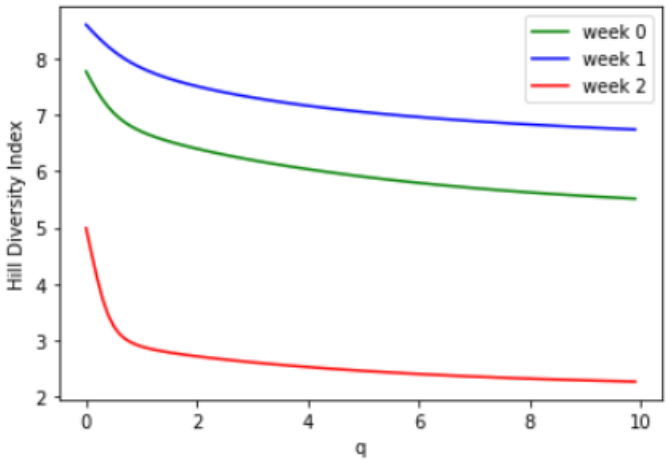


FIGURE 4.1 – Courbes d'évolution de l'indice de diversité de Hill pour les trois répertoires.

Pour cette étude de cas, nous avons considéré la colonne 'clone_id'. Nous avons donc considéré les clones et leur fréquence.

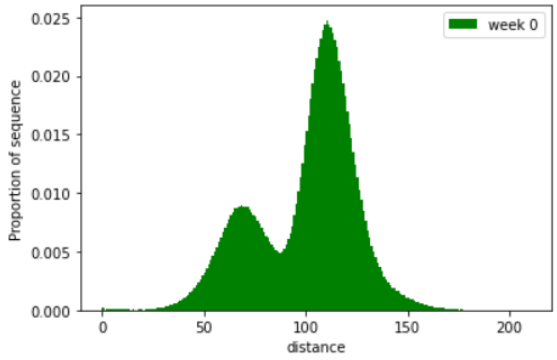
Nous remarquons que la courbe associée au répertoire de la semaine 2 est bien en-dessous de celles associées aux répertoires des semaines 0 et 1. On peut donc voir que la diversité du répertoire immunitaire commence à baisser deux semaines après la vaccination. Cependant, on remarque que les deux autres courbes sont similaires, même si la courbe associée au répertoire de la semaine 0 est en-dessous de celle du répertoire de la semaine 1. On considère que l'indice de diversité de Hill ne permet pas de distinguer ces deux répertoires.

On peut également comparer ces courbes en calculant les coefficients de corrélation de Pearson. Ces coefficients nous montrent que le répertoire de la semaine 0 est plus proche du répertoire de la semaine 1 que du répertoire de la semaine 2.

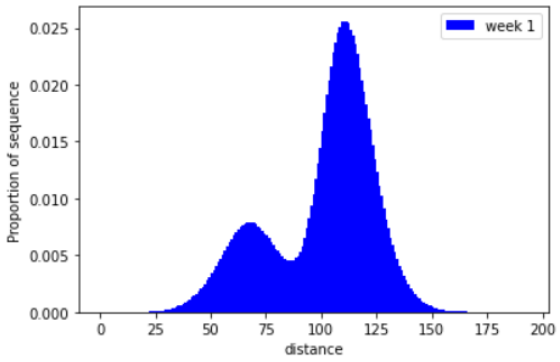
	week0	week1	week2
week0	1.000000	0.998115	0.922940
week1	0.998115	1.000000	0.909031
week2	0.922940	0.909031	1.000000

FIGURE 4.2 – Coefficients de corrélation de Pearson entre chaque paire de répertoires.

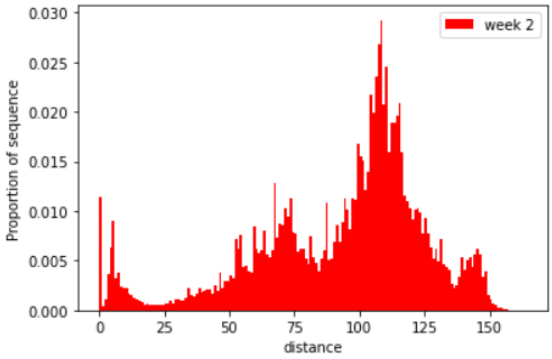
4.2 Pairwise Distance Distribution



Semaine 0



Semaine 1



Semaine 2

FIGURE 4.3 – Pairwise Distance Distribution pour les trois répertoires.

On remarque que les distributions associées aux répertoires des semaines 0 et 1 sont similaires. On peut donc supposer que ces répertoires sont similaires au niveau de la diversité.
La distribution associée au répertoire de la semaine 2 est différente des deux autres. En effet, en plus d'être irrégulière, on observe des pics à 0 et à environ 10, ce qui montre que beaucoup de séquences sont similaires ou très proches dans ce répertoire. On peut donc voir que le répertoire de la semaine 2 est moins diversifié.

On peut calculer la Jensen-Shannon Divergence entre les distributions. On voit ainsi que les trois distributions sont similaires cependant les distributions associées aux semaines 0 et 1 sont les plus proches.

	week0	week1	week2
week0	0.0	0.001819	0.034093
week1	0.0	0.000000	0.038485
week2	0.0	0.000000	0.000000

FIGURE 4.4 – Jensen-Shannon Divergence entre chaque répertoire.

4.3 Représentation en réseau

La représentation en réseau des répertoires sous forme de graphe nous donne un aperçu de leur diversité.

En effet, on voit sur la figure 4.7 que les nœuds et les composantes connexes sont de plus grande taille que ceux des deux autres graphes. Cela montre que dans le répertoire de la semaine 2, beaucoup de séquences sont identiques et que les séquences sont globalement similaires. La diversité est donc moins élevée dans ce répertoire.

Les graphes associés aux répertoires des semaines 0 et 1 (Figure 4.5 et Figure 4.6) ne présentent pas de différences remarquables. On peut donc supposer que ces répertoires sont similaires au niveau de la diversité.

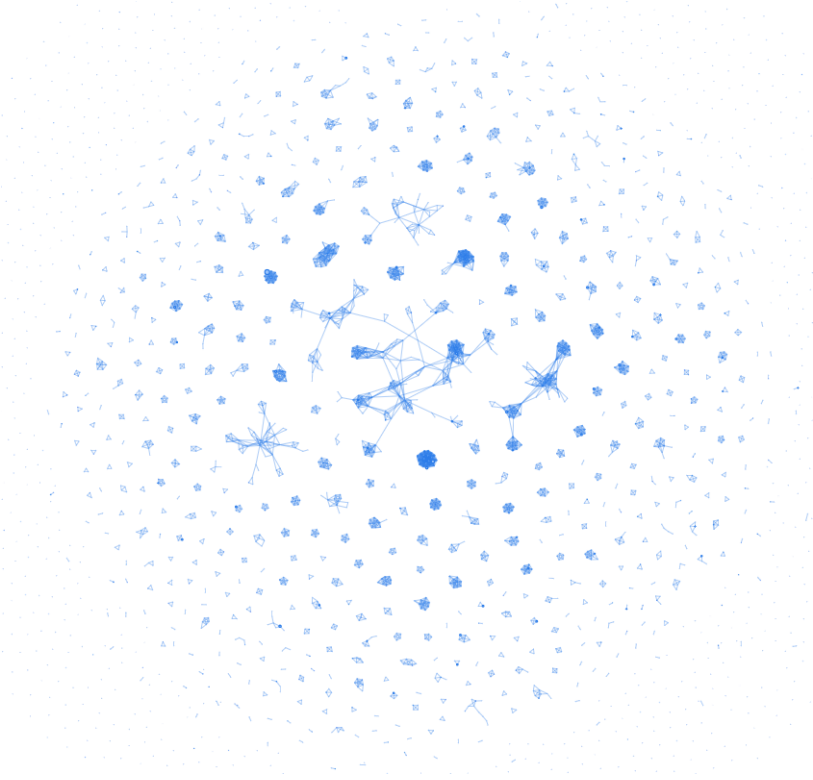


FIGURE 4.5 – Graphe représentant un échantillon de 5000 séquences du répertoire de la semaine 0

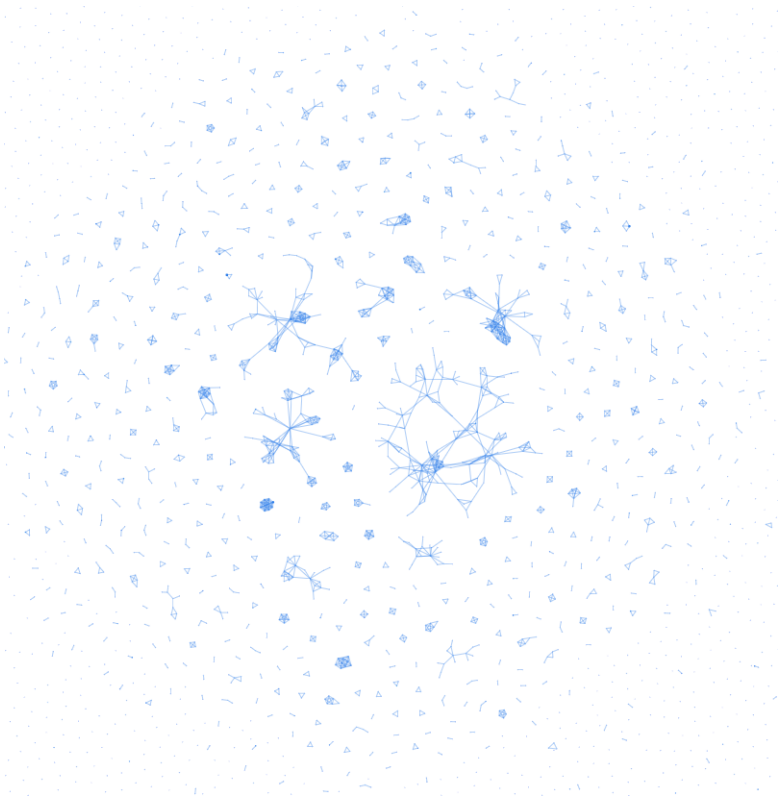


FIGURE 4.6 – Graphe représentant un échantillon de 5000 séquences du répertoire de la semaine 1

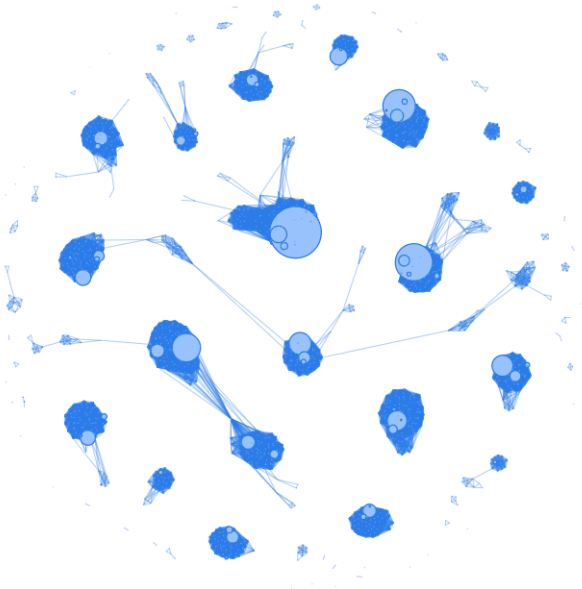


FIGURE 4.7 – Graphe représentant un échantillon de 5000 séquences du répertoire de la semaine 2

On peut également se servir des caractéristiques de ces graphes pour analyser la diversité. On remarque alors que le degré moyen des nœuds est nettement supérieur pour le graphe associé au répertoire de la semaine 2. De plus, il possède moins de composantes connexes que les deux autres graphes. Ainsi, on peut supposer que les séquences sont très similaires dans ce répertoire. Concernant les graphes des répertoires de la semaine 0 et de la semaine 1, nous considérons que leurs caractéristiques ne présentent pas assez de différences notables

pour conclure qu'il y a une différence de diversité.

	nodes	edges	average degree	nb of connected comp	max size	min size	average size
week0	4050	5315	2.62	1968	192	1	2.06
week1	4785	2482	1.04	3295	109	1	1.45
week2	1798	58664	65.25	100	217	1	17.98

FIGURE 4.8 – Caractéristiques des graphes associés aux répertoires.

4.4 Répartition des gènes IGHV et IGHJ

On peut comparer la répartition des gènes IGHV et IGHJ entre les trois répertoires.

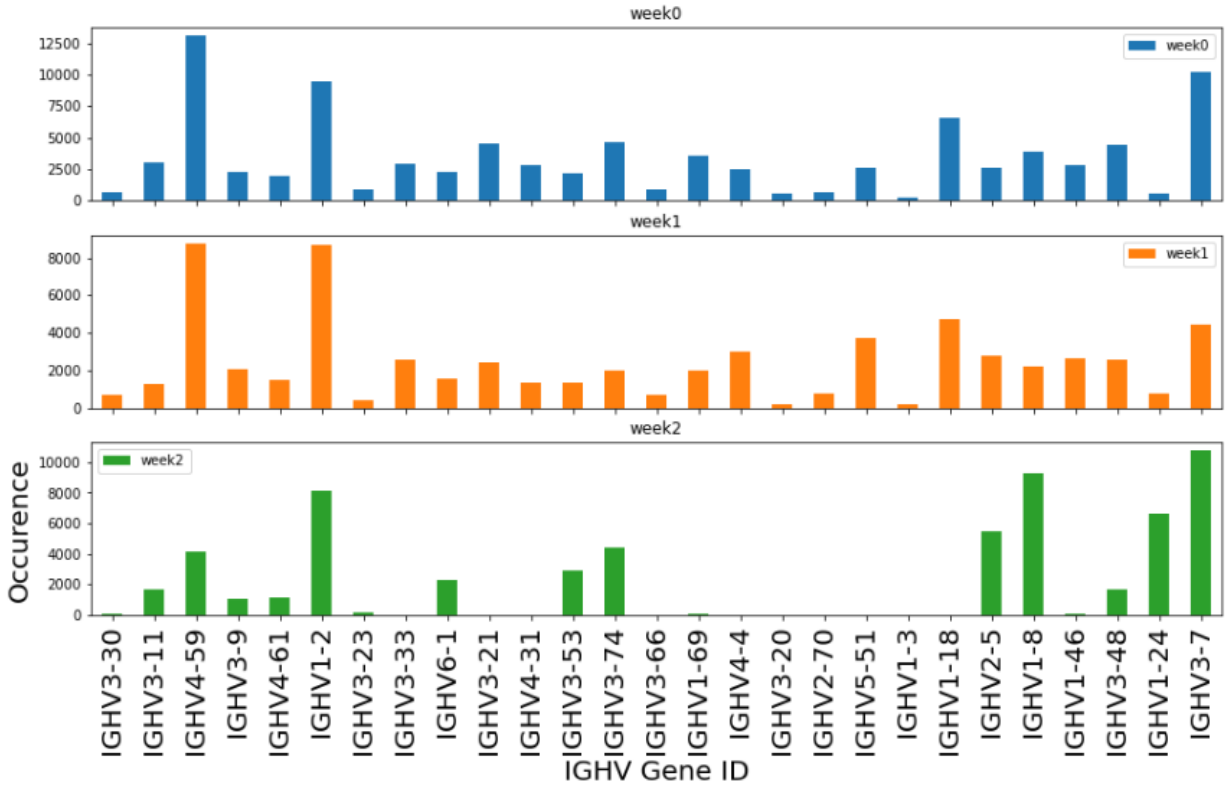


FIGURE 4.9 – Répartition des gènes IGHV en commun pour les trois répertoires.

Lors de l'affichage des histogrammes (Figures 4.9 et 4.10), nous ne prenons en compte que les gènes en commun ayant une occurrence supérieure à une certaine valeur seuil, ici 100. C'est-à-dire que si un gène possède une occurrence inférieure à 100 pour les trois répertoires, nous n'affichons pas ce gène dans l'histogramme.

On remarque que les répartitions des gènes des semaines 0 et 1 sont similaires. Elles suivent environ la même tendance. On remarque un changement lors de la semaine 2. En effet, on voit que plusieurs gènes présents dans les répertoires des semaines 0 et 1 n'apparaissent pas ou très peu dans le répertoire de la semaine 2. On observe donc un changement dans les gènes utilisés lors de la semaine 2. On peut supposer que à partir de la semaine 2, d'autres gènes sont utilisés massivement sous l'effet du vaccin, induisant une diminution de l'utilisation des gènes présents lors des semaines 0 et 1.

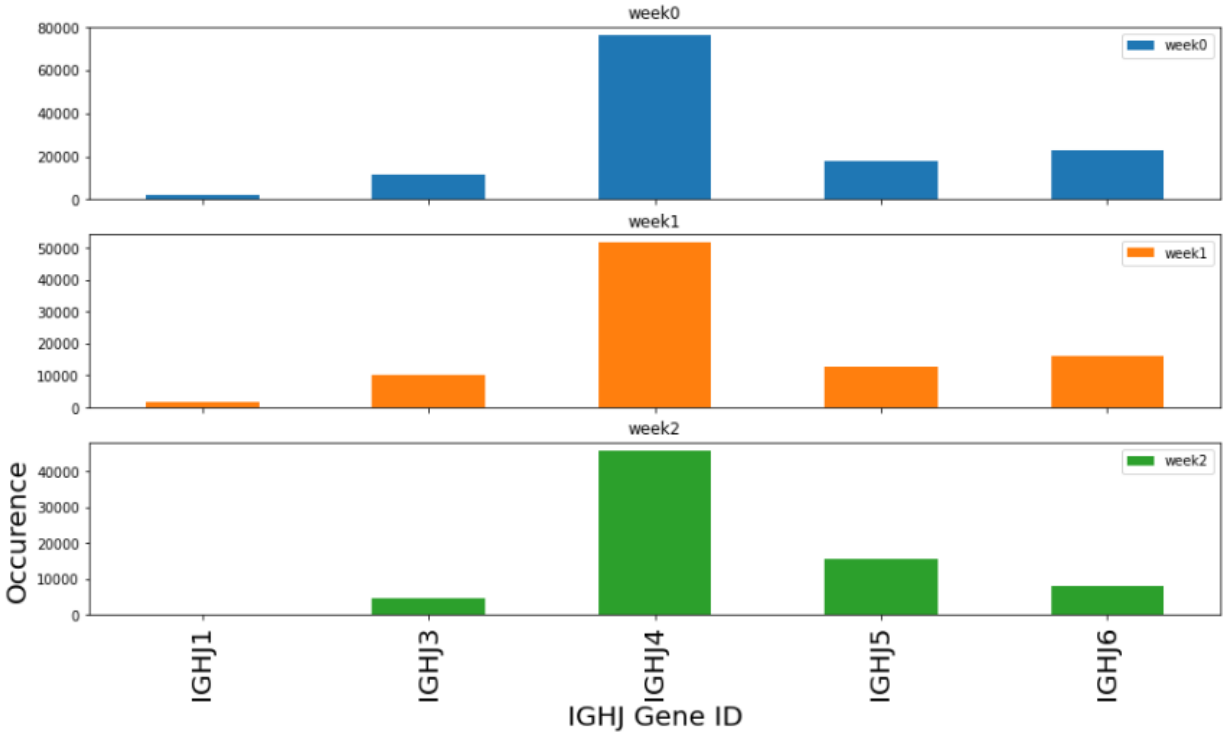


FIGURE 4.10 – Répartition des gènes IGHJ en commun pour les trois répertoires.

4.5 Pourcentage de similarité

Nous obtenons un pourcentage de similarité de 0% entre chaque répertoire. En effet, nous nous attendions à avoir un pourcentage de similarité élevé étant donné que les répertoires proviennent de la même personne. Cependant, nous avons appris que les répertoires d'un même individu peuvent contenir théoriquement jusqu'à 10^{12} séquences différentes. De ce fait, un prélèvement ne donnera qu'environ 10^5 de ces séquences, ce qui représente une petite partie de l'ensemble réel. Ainsi, si on effectue un deuxième prélèvement il y a très peu de chances de tomber sur les mêmes séquences, ce qui explique le taux de similarité de 0%.

Ainsi, grâce aux outils développés, nous avons pu déterminer que la diversité du répertoire immunitaire du patient étudié diminue à partir de la 2^e semaine après la vaccination contre la COVID-19. Cette conclusion était attendue car la vaccination fait apparaître en masse des anticorps spécifique dans le répertoire immunitaire pour se prémunir contre la maladie.
Cependant, nous nous attendions à observer une baisse de diversité directement à partir de la semaine 1. Or, les résultats obtenus ne permettent pas de conclure que le répertoire de la semaine 0 est plus diversifié que le répertoire de la semaine 1. On peut supposer que les outils dévellopés ne sont pas assez précis pour détecter une baisse de diversité. On peut aussi supposer que les répertoires des semaines 0 et 1 sont similaires voire identiques en terme de diversité et que la baisse de diversité ne s'effectue vraiment qu'à partir de la deuxième semaine.

Conclusion

Nous avons étudié et développé des outils pour quantifier et analyser la diversité au sein d'un répertoire immunitaire. Nous avons ainsi identifié plusieurs caractéristiques décrivant cette diversité et avons dévelopé les outils pour calculer et comparer ces caractéristiques entre plusieurs répertoires.
En appliquant ces outils à des données simulées, nous nous sommes assurés que l'ensemble des résultats obtenus permettent de distinguer plusieurs types de répertoires immunitaires. Nous avons ensuite appliqué ces outils à des données réelles d'un patient vacciné contre la COVID-19 et obtenu des résultats intéressants. En effet, les fonctions développées ont permis d'observer une baisse de la diversité après la vaccination.

Pour conclure, la combinaison des outils permet de distinguer des répertoires de différentes sortes et permet également d'observer l'évolution d'un même répertoire immunitaire au cours du temps. Cependant, des améliorations restent à effectuer, notamment pour rendre les outils plus précis, plus lisibles et plus rapides.

Ouverture

Visualisation des graphes

Un des problèmes rencontré concerne la visualisation des graphes. En effet, nous utilisons la bibliotèque Python Pyvis pour afficher les graphes dans une fenêntre HTML. Cependant, le temps d'affichage du graphe est égal voire supérieur au temps de calcul des distances entre les nœuds. On arrive alors à un temps d'éxécution total d'environ 5 minutes pour un graphe de 5000 nœuds.

Nous avons donc tenté d'utiliser l'outil d3.js qui permet d'afficher des graphes en passant par du JavaScript. Cependant par manque de temps pour nous familiariser avec ce langage, nous n'avons pas pu appliquer cette amélioration. Nous avons néanmoins créer une fonction pour convertir nos graphes en format .json accepté par l'outil d3.js pour faciliter la lecture de ceux-ci dans de futurs travaux.

Création d'un classifieur de répertoire

Une des idées d'extensions pour ce projet est la création d'un classifieur capable de distinguer les différents types de répertoires en fonction de leur diversité.

Cela pourrait s'effectuer par le biais d'un réseau de neurones qui prendrait en entrée les résultats obtenus avec les outils qui ont été développés, et donnerait en sortie la classe du répertoire considéré : monoclonal, polyclonal ou oligoclonal. Un réseau adéquat à cette tâche posséderait une ou plusieurs couches de neurones intermédiaires. Nous pourrions effectuer l'apprentissage de manière supervisé avec un algorithme de retropropagation cependant cela nécessite beaucoup de données annotées. On peut alors penser à effectuer l'apprentissage de façon non supervisé pour éviter d'annoter les répertoires de la base de données.

Bibliographie

[1] Bolen, C. R., Rubelt, F., Vander Heiden, J. A., and Davis, M. M. The Repertoire Dissimilarity Index as a method to compare lymphocyte receptor repertoires. *BMC Bioinformatics 18*, 1 (Mar. 2017), 155.

[2] Laydon, D. J., Bangham, C. R. M., and Asquith, B. Estimating t-cell repertoire diversity : limitations of classical estimators and a new approach. *Philosophical Transactions of the Royal Society B : Biological Sciences 370* (2015).

[3] Laydon, D. J., Melamed, A., Sim, A., Gillet, N. A., Sim, K., Darko, S., Kroll, J. S., Douek, D. C., Price, D. A., Bangham, C. R. M., and Asquith, B. Quantification of htlv-1 clonality and tcr diversity. *PLOS Computational Biology 10*, 6 (06 2014), 1–13.

[4] Olson, B. J., Moghimi, P., Schramm, C. A., Obraztsova, A., Ralph, D., Vander Heiden, J. A., Shugay, M., Shepherd, A. J., Lees, W., and Matsen, F. A. sumrep : A summary statistic framework for immune receptor repertoire comparison and model validation. *Frontiers in Immunology 10* (2019).

[5] Weber, C. R., Rubio, T., Wang, L., Zhang, W., Robert, P. A., Akbar, R., Snapkov, I., Wu, J., Kuijjer, M. L., Tarazona, S., Conesa, A., Sandve, G. K., Liu, X., Reddy, S. T., and Greiff, V. Reference-based comparison of adaptive immune receptor repertoires. *bioRxiv* (2022).