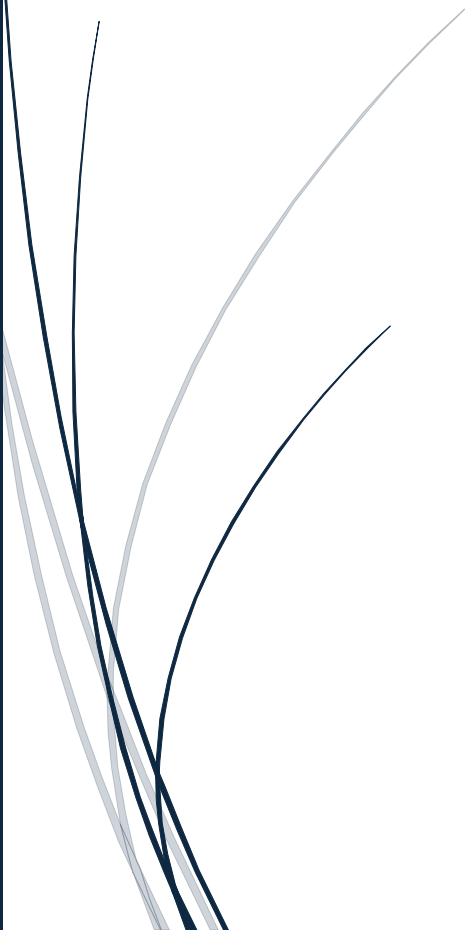




2024/2025

# CHPS1001

## Analyse de la structure des membranes lipidiques



Simao CORTINHAL, Romain COGNÉ, Quentin BATISTE,  
Zinedine BOUZÉKKAR, Gael ROUBAHIE-FISSA  
MASTER CHPS 2

## Sommaire

|  |    |
|--|----|
| Introduction .....   | 1  |
| Matériels et méthodes .....  | 2  |
| Résultats et interprétations.....  | 4  |
| Première version du code – seuil de distance automatisé .....  | 4  |
| Deuxième version du code – Centre de masse sans atome de référence.....                                | 5  |
| Troisième version du code – Orientation lipides et calcul normales locales avec seuil automatisé ..... | 7  |
| Conclusion .....   | 9  |
| Table des annexes.....   | 10 |
| Annexes.....   | 11 |

## Introduction

Dans le domaine de la bio-informatique, la modélisation des membranes lipidiques est importante pour mieux comprendre les processus cellulaires. Ces membranes, constituées de bicouches lipidiques, jouent un rôle important dans le maintien de l'intégrité des cellules et la régulation de divers mécanismes biologiques. La simulation informatique de ces structures permet de progresser dans l'analyse de leurs propriétés. Cependant, elle pose également des défis en raison de la complexité toujours plus importante des systèmes étudiés.

Ce projet a pour but de concevoir un programme capable de classer les lipides de chaque feuillet d'une bicouche membranaire dans une couche ou l'autre. En adoptant des approches basées sur la théorie des graphes, la normale locale ou l'orientation des lipides, le but du projet est de proposer une méthodologie optimisée pour surmonter les limitations des techniques actuelles.

## Matériels et méthodes

Pour résoudre le problème posé, nous avons principalement utilisé le langage Python en combinaison avec plusieurs bibliothèques spécialisées. On a choisi ce langage pour sa facilité, mais surtout pour son large choix de bibliothèques, notamment pour la bio-informatique, ce qui nous a évité de devoir programmer par nous-mêmes certaines fonctions qui nous auraient fait perdre un certain temps.

Les données fournies étaient des fichiers de structure au format GRO, contenant les positions des lipides dans les membranes. Ces fichiers ont été chargés et analysés à l'aide de la bibliothèque **MDAnalysis**.

Pour analyser les membranes, nous avons utilisé la théorie des graphes. La bibliothèque **NetworkX** a été utilisée pour représenter les membranes sous forme de graphes, où chaque lipide est un sommet et les arêtes sont définies par une distance entre les atomes de référence. La construction et l'analyse de ces graphes ont permis d'assigner chaque lipide à un feuillet de la membrane.

La visualisation des membranes et des résultats intermédiaires a été effectuée de deux manières. Tout d'abord, Matplotlib nous a permis de vérifier rapidement nos résultats dans un premier temps. **VMD** a permis quant à lui de visualiser directement les structures et de vérifier l'intégrité des données fournies.

Enfin, plusieurs étapes ont été nécessaires pour implémenter les différentes versions du programme. Nous avons d'abord développé une méthode utilisant un seuil de distance automatisé pour construire un graphe simple, qui permet donc de classer les lipides des feuillet. Ensuite, une deuxième version a été mise en place, en faisant les centres de masses des lipides pour permettre de ne plus sélectionner d'atome de référence. Une troisième version a intégré le calcul des normales locales pour prendre en compte les orientations des lipides dans les membranes. À chaque étape, le code a été optimisé pour réduire les temps de calcul et garantir une fiabilité maximale des résultats.

Pour toutes les versions, on utilise un seuil de distance adaptatif, les lipides sont sélectionnés automatiquement en excluant les résidus non lipidiques (e.g., "W", "NA+", "CL-", etc). On récupère aussi les dimensions de la boîte pour appliquer les conditions périodiques aux calculs (PBC : periodic boundary conditions).

Cela assure que les relations entre lipides sont calculées de manière réaliste, comme si la membrane s'étendait infiniment dans toutes les directions, les lipides proches des bords sont donc bien calculés ici.

Au départ, pour la **première méthode**, nous avons une approche utilisant uniquement le CPU. Cette approche avait tendance à être extrêmement longue pour les fichiers small et large plasma membrane. Donc, nous avons essayé d'accélérer

l'exécution en parallélisant le code. Cependant, l'exécution étant toujours trop longue sur les plus gros fichiers, nous avons donc décidé de paralléliser sur GPU à l'aide de CUDA (Numba en python) car chaque calcul étant indépendant, cela accélère l'établissement de la matrice des distances.

Pour construire le graphe qui regroupe les lipides en feuillet, on a mis en place la fonction « *find\_leaflets\_by\_graph* ». Ici, chaque lipide est représenté comme un nœud et des arêtes sont ajoutées entre les lipides dont la distance est inférieure au seuil. Les composantes connexes correspondent aux feuillets membranaires. On ajuste ensuite le seuil si le graphe contient plus de deux composantes, ce qui fusionne les petites composantes avec les grandes sans fusionner les grandes.

Pour la **deuxième méthode**, afin de ne pas définir d'atome de référence, nous avons alors décidé dans un premier temps de nous baser sur la moyenne des coordonnées de chaque lipide dans chaque axe. À partir de ces coordonnées, nous nous servons de la méthode des graphes comme dans la méthode précédente pour déterminer la position de ce lipide dans le feuillet. Il s'avère que les résultats sur les vésicules étaient peu concluants. Nous avons alors voulu effectuer le calcul de la médiane des coordonnées, mais là aussi cette méthode ne résout pas notre problème.

Nous avons alors essayé d'utiliser le centre de masse qui est le point de l'espace où l'on applique les effets d'inertie. Pour calculer celui-ci, on extrait les positions et les masses de chaque atome, on somme les poids pour obtenir la masse du lipide. On multiplie alors chaque position de chaque lipide par sa masse afin d'obtenir les positions pondérées par la masse. On extrait alors les coordonnées obtenues et on normalise pour obtenir ce centre de masse donnant de meilleurs résultats. La méthode des graphes est toujours utilisée pour regrouper les lipides en feuillets.

Enfin, la **dernière méthode** concerne la version qui ajoute l'orientation des lipides et le calcul de la normale locale à chaque lipide, mais également la version sur base des éléments nécessaires et suffisants.

Les normales sont calculées à partir des coordonnées des atomes de chaque lipide et de leurs voisins proches, en utilisant un seuil de distance. Pour chaque lipide, un vecteur normal local a été défini comme la moyenne des vecteurs reliant le centre géométrique du lipide à celui de ses voisins.

L'orientation des lipides a été déterminée en sélectionnant deux atomes représentatifs (« tête et queue ») pour chaque lipide. Le vecteur reliant ces deux atomes a été normalisé et utilisé pour calculer des similarités cosinus entre les lipides, ce qui offre un critère supplémentaire pour différencier les feuillets. Cette méthode utilise les normales locales et les orientations des lipides calculées. Les nœuds (lipides) du graphe sont connectés par des arêtes si deux conditions sont remplies :

- La distance entre les lipides est inférieure à un seuil défini.
- La similarité d'orientation (cosinus) ou l'alignement des normales locales dépassent un seuil donné.

Lors de la construction du graphe, on garde l'automatisation des seuils de distance présentés dans les méthodes précédentes.

On a utilisé plusieurs techniques pour réduire le temps de calcul. D'abord les distances entre les lipides ont été calculées en tenant compte des conditions périodiques à l'aide de la bibliothèque CUDA avec Numba afin d'accélérer les opérations sur GPU. Enfin, la matrice de similarité d'orientation a été parallélisée sur GPU pour des calculs rapides sur de grands ensembles de données.

## Résultats et interprétations

Ce qu'il faut savoir, c'est que nous avons à disposition des fichiers de résultats fournis avec les fichiers.gro des membranes. Ces fichiers textes de résultats contiennent la classification de chacun des lipides dans l'un ou l'autre feuillet en utilisant leurs numéros associés dans les fichiers.gro. Nous devons donc les utiliser en comparant nos résultats avec ceux fournis.

### Première version du code – seuil de distance automatisé

#### Résultats :

Pour la première version du code, utilisant un seuil de distance pour construire un graphe, nous avons obtenu des résultats plutôt similaires aux fichiers fournis :

- **Aucune** différence pour : *bilayer\_peptide*, *bilayer\_prot*, *model\_vesicle*
- **3** différences pour : *bilayer\_chol*
- Entre **15** et **20** différences : *large\_plasma\_membrane*, *small\_plasma\_membrane*
- Ainsi qu'un autre cas pour *dppc\_vesicle* (40 additions de lipides).

Pour les affichages, voir les figures 2 à 8 dans l'annexe.

| Nom du fichier .gro   | Nombre de lipides | Pourcentage d'exactitude avec les fichiers de sortie |
|-----------------------|-------------------|--|
| Bilayer_chol          | 1944              | 99,85  |
| Bilayer_peptide       | 126               | 100,00   |
| Bilayer_prot          | 112               | 100,00   |
| Dppc_vesicle          | 3072              | 100,00   |
| Large_plasma_membrane | 19239             | 99,78  |
| Small_plasma_membrane | 6664              | 99,77  |
| Model_vesicle         | 2748              | 100,00   |

### Interprétations :

En premier lieu, cette méthode se montre très efficace sur les petits fichiers, où les lipides sont peu nombreux et bien espacés. En effet, on ne remarque aucune différence entre les fichiers de résultats, ce qui peut être expliqué par le fait qu'il est assez aisé de différencier les feuillettes. On remarque cependant quelques différences sur les plus gros fichiers de résultats, où le nombre de lipides au centre de la membrane est plus important.

Également, le cas de *dppc\_vesicle* est assez différent, car on pense que la méthode utilisée pour générer les fichiers fournis de résultats (avec le sujet) ne prend pas en compte certains lipides. On pourrait dire qu'ils ont été « oubliés ». En effet nous n'avons aucune suppression, seulement 40 additions de lipides. Ainsi, on n'aurait aucune différence pour ce fichier également.

Les fichiers contenant un petit nombre de lipides, tels que *bilayer\_peptide*, *bilayer\_prot* et *model\_vesicle*, présentent une correspondance parfaite (100% d'exactitude) avec les fichiers de référence. Cela peut s'expliquer avec la simplicité de ces systèmes, où les lipides sont bien répartis et clairement séparés entre les deux feuillettes.

Le fichier *bilayer\_chol*, avec un pourcentage d'exactitude de 99,85 %, montre quelques petites divergences. Même si le nombre de lipides est inférieur au fichier *Model\_vesicle*, il se peut que la disposition des lipides ainsi que la structure générale de la membrane empêchent la méthode d'obtenir 100% d'exactitude.

Pour les fichiers *large\_plasma\_membrane* et *small\_plasma\_membrane*, où les lipides sont nombreux (respectivement 19239 et 6664), le pourcentage d'exactitude reste élevé (99,78 % et 99,77 %). Ces résultats montrent que la méthode est robuste même pour les membranes complexes.

### Deuxième version du code – Centre de masse sans atome de référence

#### Résultats :

Pour la deuxième version du code, utilisant le centre de masse des lipides sans définition d'atomes de références, nous avons obtenu des résultats plutôt similaires aux fichiers fournis :

- **Aucune** différence pour : *bilayer\_peptide*, *bilayer\_prot*, *model\_vesicle*
- **6** différences pour : *bilayer\_chol*
- Pas pu déterminer les différences : *large\_plasma\_membrane*
- **565** différences : *small\_plasma\_membrane*
- **295** différences : *dppc\_vesicle*

Ainsi qu'un autre cas pour *dppc\_vesicle* (40 additions de lipides).

Pour les affichages, voir les figures 9 à 15 dans l'annexe.

| Nom du fichier .gro   | Nombre de lipides | Pourcentage d'exactitude avec les fichiers de sortie |
|-----------------------|-------------------|--|
| Bilayer_chol          | 1944              | 99,69  |
| Bilayer_peptide       | 126               | 100,00   |
| Bilayer_prot          | 112               | 100,00   |
| Dppc_vesicle          | 3072              | 90,39  |
| Large_plasma_membrane | 19239             | X  |
| Small_plasma_membrane | 6664              | 91,52  |
| Model_vesicle         | 2748              | 100,00   |

### Interprétations :

Pour les fichiers bilayer\_peptide, bilayer\_prot et model\_vesicle, on obtient de très bons résultats. Cette méthode se montre très efficace sur les petits fichiers, où les lipides sont peu nombreux et bien espacés. En effet, on ne remarque aucune différence entre les fichiers de résultats, ce qui peut être expliqué par le fait qu'il est assez aisé de distinguer les feuillettes. Une différence de 6 lipides sur Bilayer\_chol peut être constatée, car il se peut que la disposition des lipides ainsi que la structure générale de la membrane empêchent la méthode d'obtenir 100% d'exactitude.

De grandes différences sont constatées sur dppc\_vesicle, et small\_plasma\_membrane. Vis-à-vis de dppc\_vesicle, sa forme complique la distinction claire des feuillettes. Pour small\_plasma\_membrane, le calcul du centre de masse pour les lipides se situant entre les 2 feuillettes sont beaucoup plus difficiles à placer que les autres.

Pour large\_plasma\_membrane, il n'a pas été possible de réaliser les mesures n'ayant pas réussi à réduire le temps d'exécution qui était très grand.



### Troisième version du code – Orientation lipides et calcul normales locales avec seuil automatisé

#### Résultats :

Pour la troisième version du code, utilisant l'orientation des lipides, ainsi que les calculs des normales locales, nous avons obtenu des résultats plutôt similaires aux fichiers fournis :

- **Aucune** différence pour : *bilayer\_peptide*, *bilayer\_prot* , *model\_vesicle*
- 2 différences pour : *bilayer\_chol*
- 5 différences pour : *dppc\_vesicle*
- 10 différences pour : *small\_plasma\_membrane* (en réduisant le pas d'augmentation du seuil de distance pour une précision plus fine)
- ~36 différences pour : *large\_plasma\_membrane*
- Ainsi qu'un autre cas pour *dppc\_vesicle* (40 additions de lipides).

Pour les affichages, voir les figures 16 à 23 dans l'annexe.

| Nom du fichier .gro   | Nombre de lipides | Pourcentage d'exactitude avec les fichiers de sortie |
|-----------------------|-------------------|--|
| Bilayer_chol          | 1944              | 99.89  |
| Bilayer_peptide       | 126               | 100,00   |
| Bilayer_prot          | 112               | 100,00   |
| Dppc_vesicle          | 3072              | 99.83  |
| Large_plasma_membrane | 19239             | 99.82  |
| Small_plasma_membrane | 6664              | 99.8   |
| Model_vesicle         | 2748              | 100,00   |

#### Interprétations :

Lors de l'analyse de l'orientation et des vecteurs normaux locaux des lipides sur la base de leurs voisins les plus proches (en utilisant la matrice de distance précalculée), nous observons de légères améliorations en termes de précision et de rapidité par rapport à la version à seuil automatique.

Une observation clé des graphiques présentés dans ce rapport est que les lipides d'un même feuillet ont souvent des orientations ou des directions similaires. Cela suggère que l'orientation devrait idéalement être le paramètre principal pour filtrer les connexions entre les lipides. Cependant, les résultats empiriques révèlent que ce n'est pas toujours le cas (surtout dans le cas de faux positifs où les atomes voisins dans le même feuillet n'ont pas la même orientation). Par conséquent, nous nous appuyons toujours sur le paramètre de distance comme premier filtre, l'orientation et les vecteurs

normaux locaux servant de critères secondaires pour confirmer l'appartenance de deux lipides à un même feuillet.

De plus, on observe des limites avec la similitude par cosinus, dans lequel la distribution de ces valeurs est quasi-uniforme comme vu dans la figure suivante :

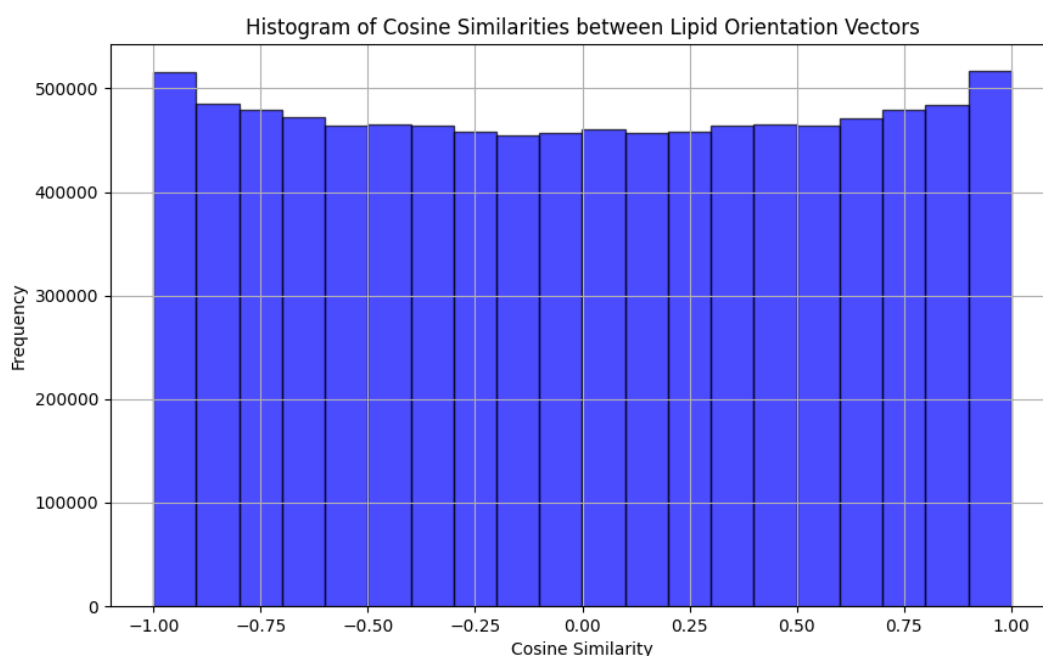


Figure 1. Histogramme de la similitude des orientations de lipides par cosinus pour le fichier *dppc\_vesicule*

Malgré ces limitations, cette méthode démontre une grande sensibilité au seuil d'orientation. Une version adaptative du seuil permettrait probablement d'obtenir des performances encore meilleures en s'ajustant dynamiquement aux caractéristiques des différentes membranes. Une configuration plus spécifique pour le fichier *large\_plasma\_membrane* est nécessaire et est disponible dans le script 'membrane-analyzer-large.py'.

Néanmoins, la méthode atteint d'excellentes performances sur toutes les membranes testées, fournissant des assignations précises de lipides à des feuillets et démontrant son efficacité pratique.

## Conclusion

Pour conclure, ce projet nous a permis de concevoir et d'implémenter des outils d'analyse de membranes lipidiques. Trois approches distinctes ont été développées. La première version, basée sur un seuil de distance automatisé, a démontré une grande robustesse et efficacité pour les membranes simples et de taille modérée. Cependant, ces limites sont apparues pour des structures avec des placements complexes.

La deuxième version, quant à elle, a exploré l'introduction du centre de masse. Bien que cette méthode ait offert des résultats intéressants, son exactitude s'est avérée inférieure dans certains cas.

Enfin, la troisième version a marqué une amélioration en intégrant les orientations des lipides et le calcul des normales locales, combinés à une optimisation des seuils. Cette méthode s'est révélée particulièrement efficace pour traiter des systèmes complexes tout en réduisant les erreurs de classification.

L'ensemble des travaux a montré l'importance d'une approche combinant critères de distance et d'orientation pour analyser plus efficacement les feuillettes. De plus, l'utilisation de la parallélisation GPU a considérablement accéléré les calculs, ce qui rend ces méthodes plus adaptées à une utilisation sur de grands ensembles.

## Table des annexes

|   |    |
|---|----|
| Figure 1. Histogramme de la similitude des orientations de lipides par cosinus pour le fichier dppc_vesicle.....  | 8  |
| Figure 2. Résultat Seuil Auto : Bilayer_chol .....  | 11 |
| Figure 3. Résultat Seuil Auto : Bilayer_peptide .....   | 12 |
| Figure 4. Résultat seuil auto : bilayer_prot .....  | 13 |
| Figure 5. Resultat Seuil Auto : Small_plasma_membrane .....   | 16 |
| Figure 6. Résultat seuil auto : model_vesicle .....   | 17 |
| Figure 7. Résultat Centre de masse : bilayer_chol.....  | 18 |
| Figure 8. Résultat Centre de masse : bilayer_peptide .....  | 19 |
| Figure 9. Résultat Centre de masse : bilayer_prot .....   | 20 |
| Figure 10. Résultat Centre de masse : dppc_vesicle .....  | 21 |
| Figure 11: Résultat Centre de masse : large_plasma_membrane .....   | 22 |
| Figure 12. Résultat Centre de masse : small_plasma_membrane .....   | 23 |
| Figure 13. Résultat Centre de masse : model_vesicle.....  | 24 |
| Figure 14. Résultat orientation vecteur normale : Bilayer Chol .....  | 25 |
| Figure 15. Résultat orientation vecteur normale : Bilayer Peptide.....  | 26 |
| Figure 16. Résultat orientation vecteur normale : Bilayer Proteine.....   | 26 |
| Figure 17. Résultat orientation vecteur normale : Zoom sur la partie droite de la membrane de DPPC Vesicule ..... | 27 |
| Figure 18. Résultat orientation vecteur normale : Vesicule Model .....  | 28 |
| Figure 19. Résultat orientation sans vecteur normale local : Small Plasma Membrane                                | 29 |
| Figure 20. Résultat orientation avec vecteur normale local : Small Plasma Membrane                                | 30 |
| Figure 21. Résultat orientation sans vecteur normale local : Large Plasma Membrane                                | 31 |

## Annexes

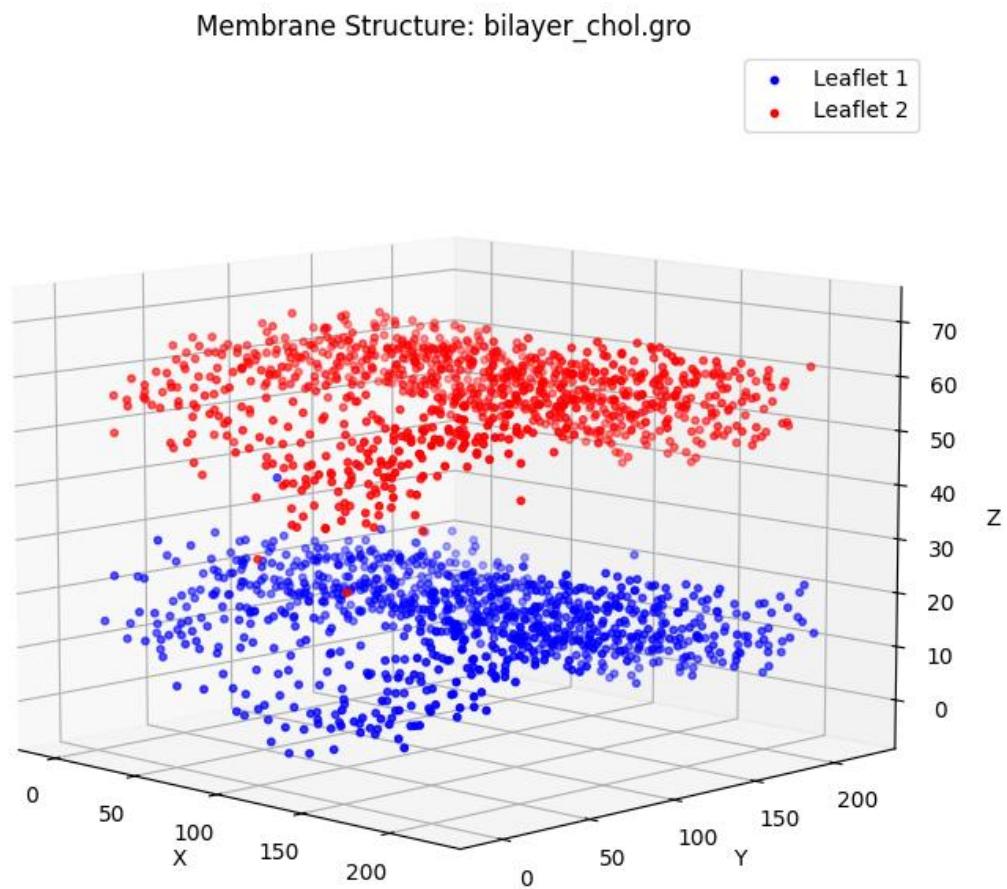


Figure 2. Résultat Seuil Auto : Bilayer\_chol

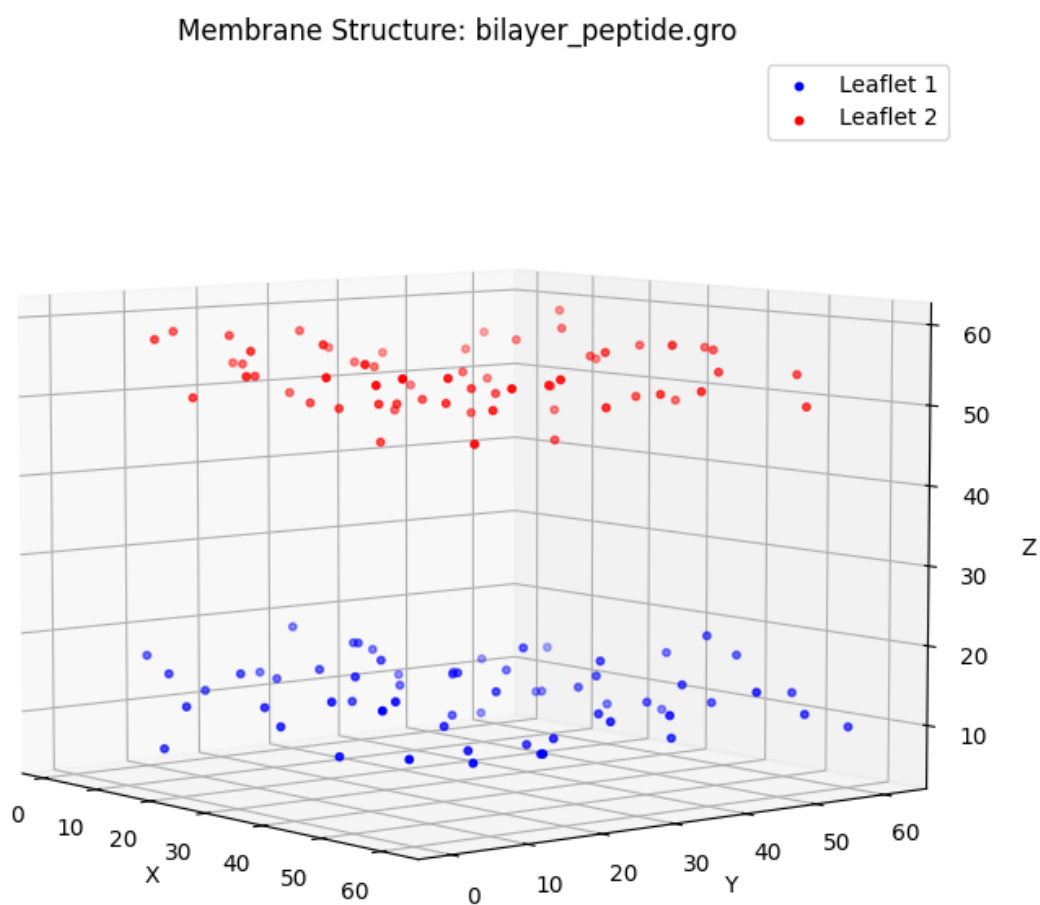


Figure 3. Résultat Seuil Auto : Bilayer\_peptide

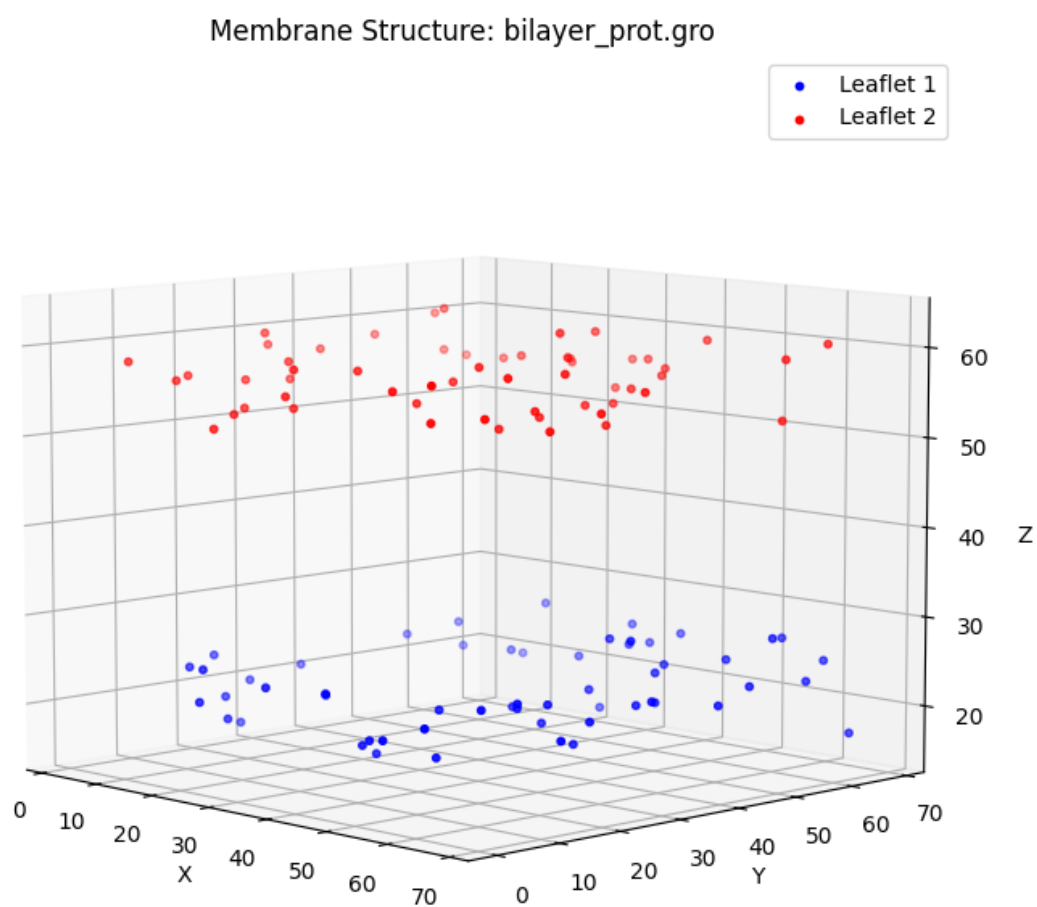


Figure 4. Résultat seuil auto : bilayer\_prot

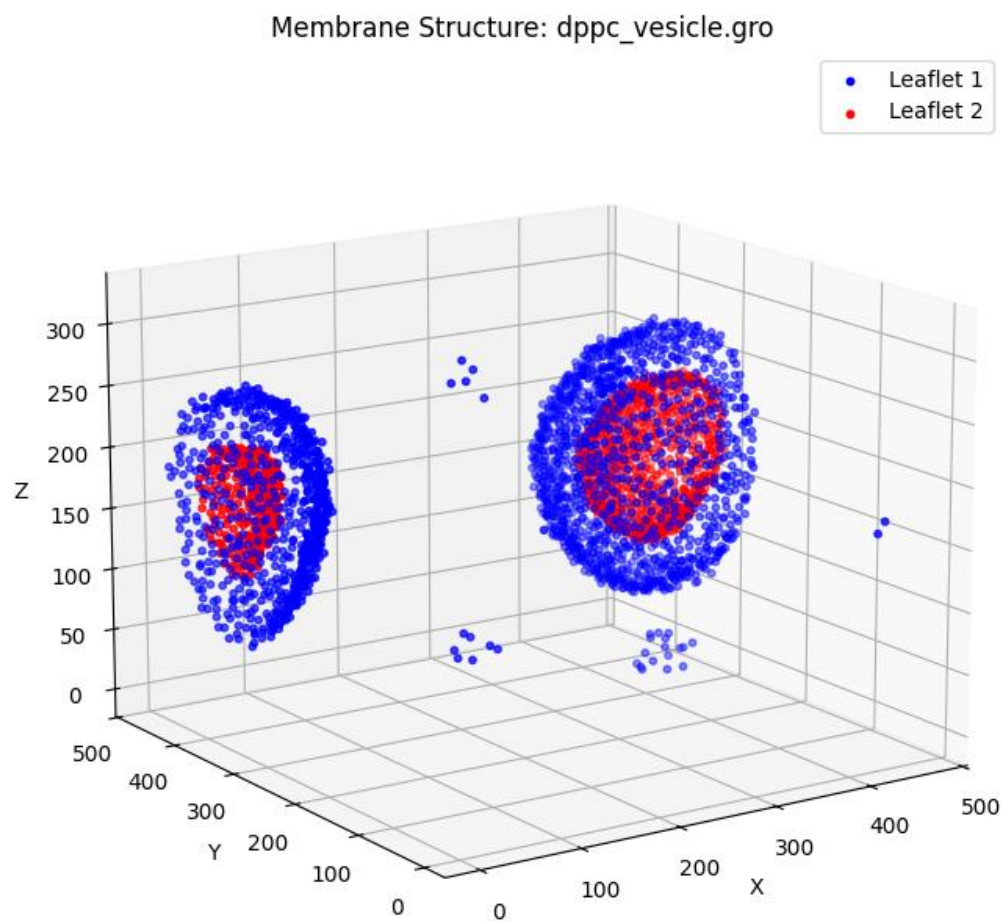


Figure 5. Résultat seuil auto : dppc\_vesicle



Membrane Structure: large\_plasma\_membrane.gro

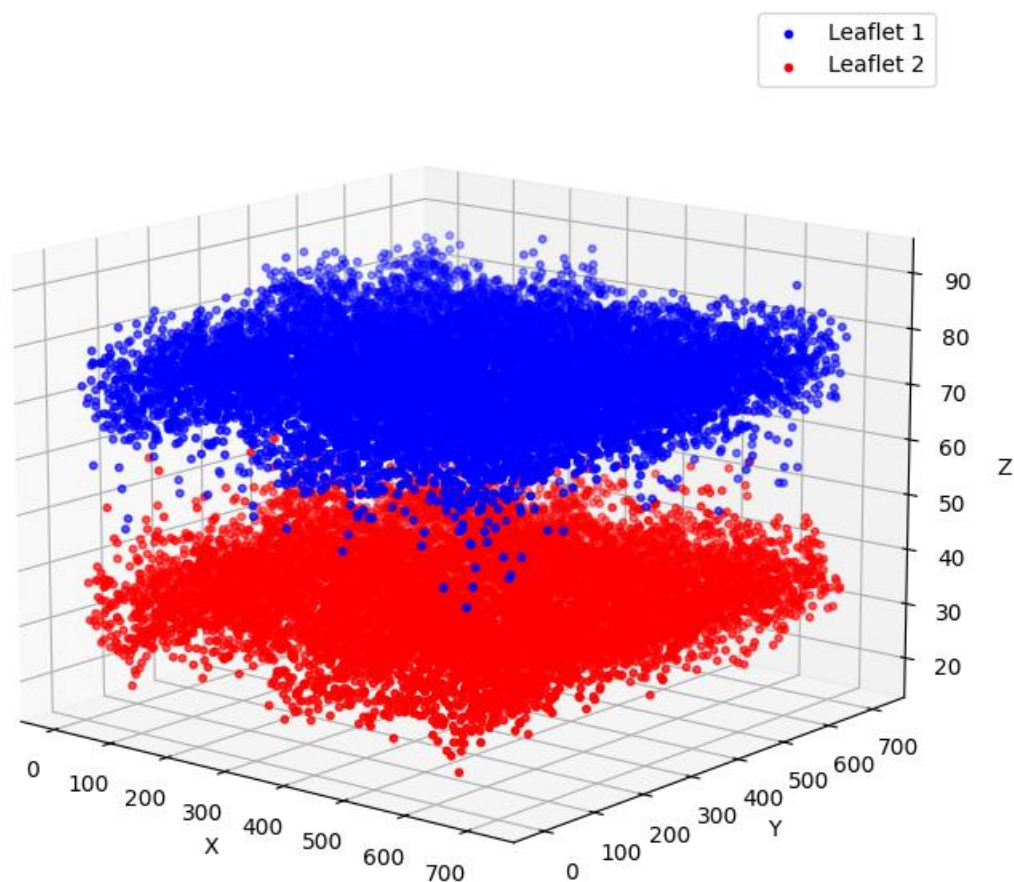


Figure 6. Résultat Seuil auto : large\_plasma\_membrane

Membrane Structure: small\_plasma\_membrane.gro

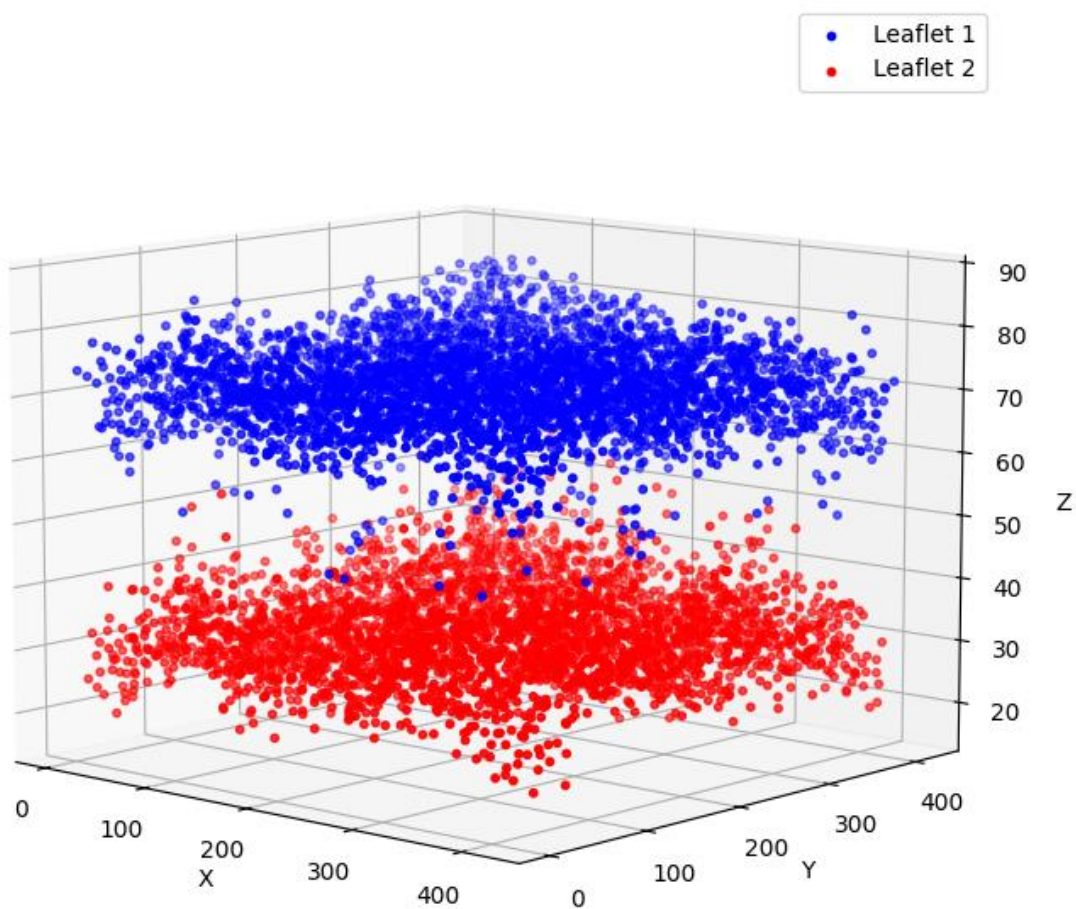


Figure 5. Resultat Seuil Auto : Small\_plasma\_membrane

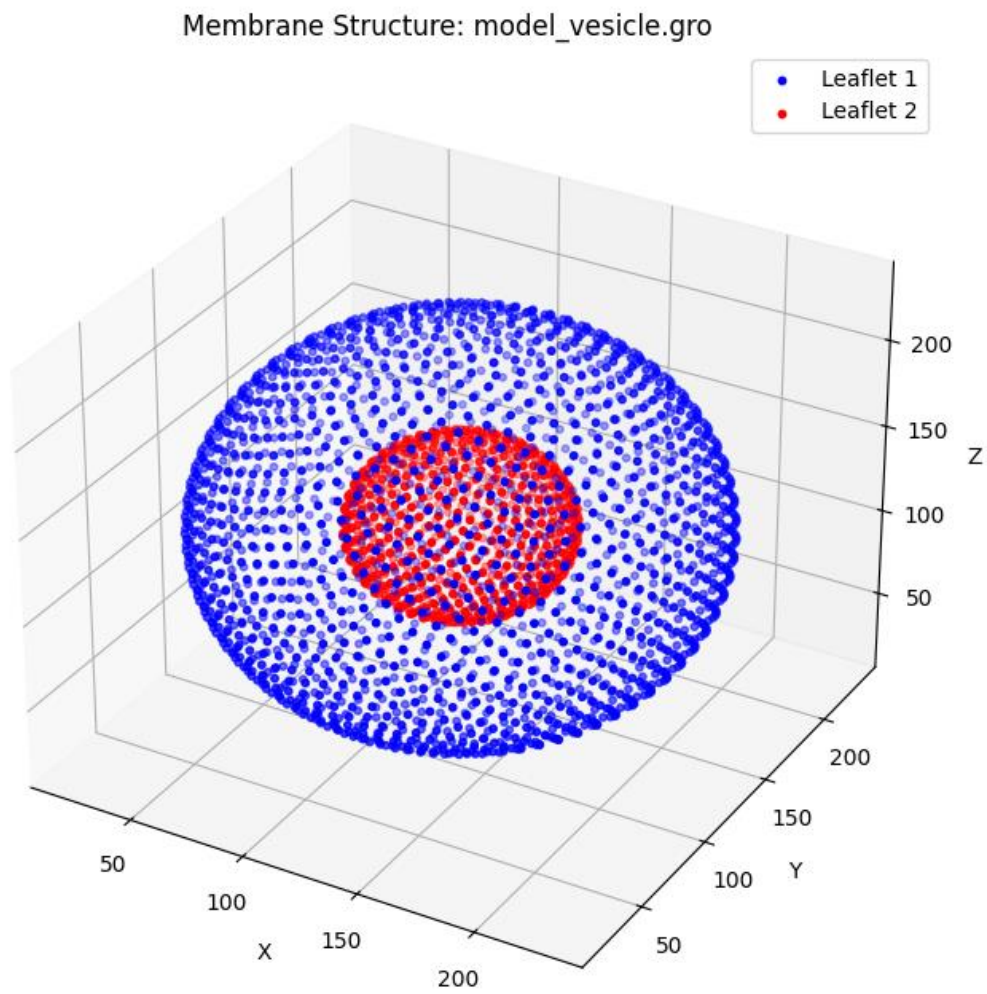


Figure 6. Résultat seuil auto : model\_vesicle

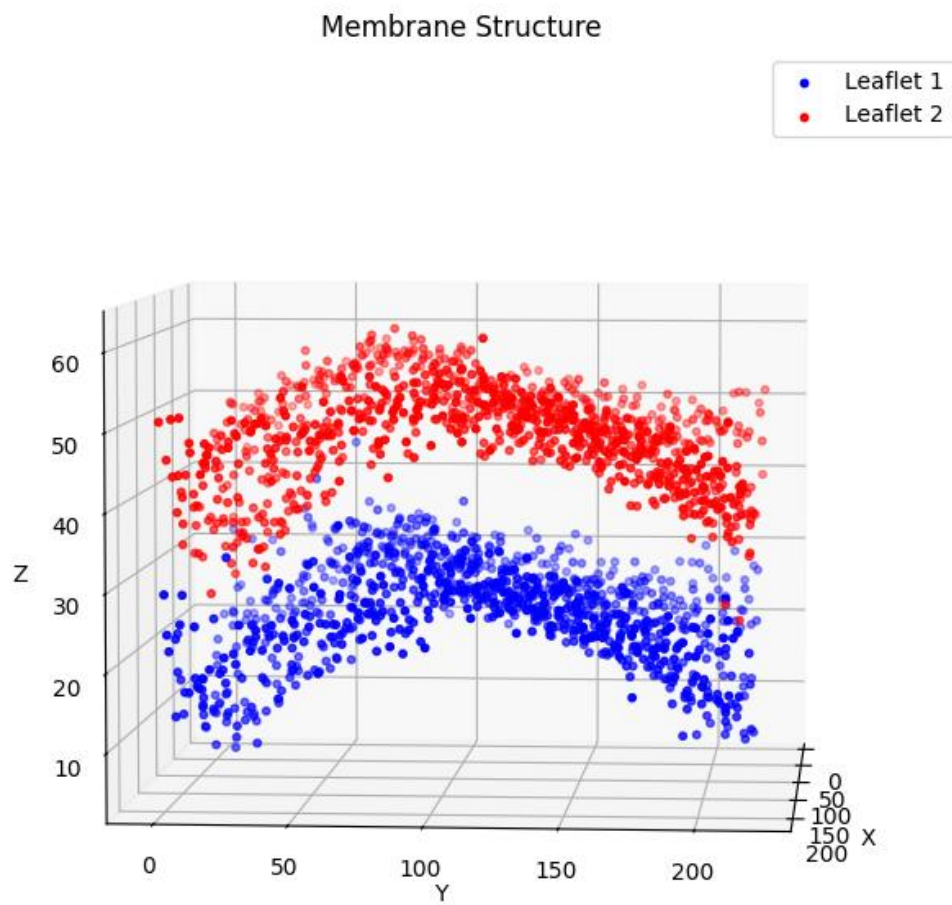


Figure 7. Résultat Centre de masse : bilayer\_chol

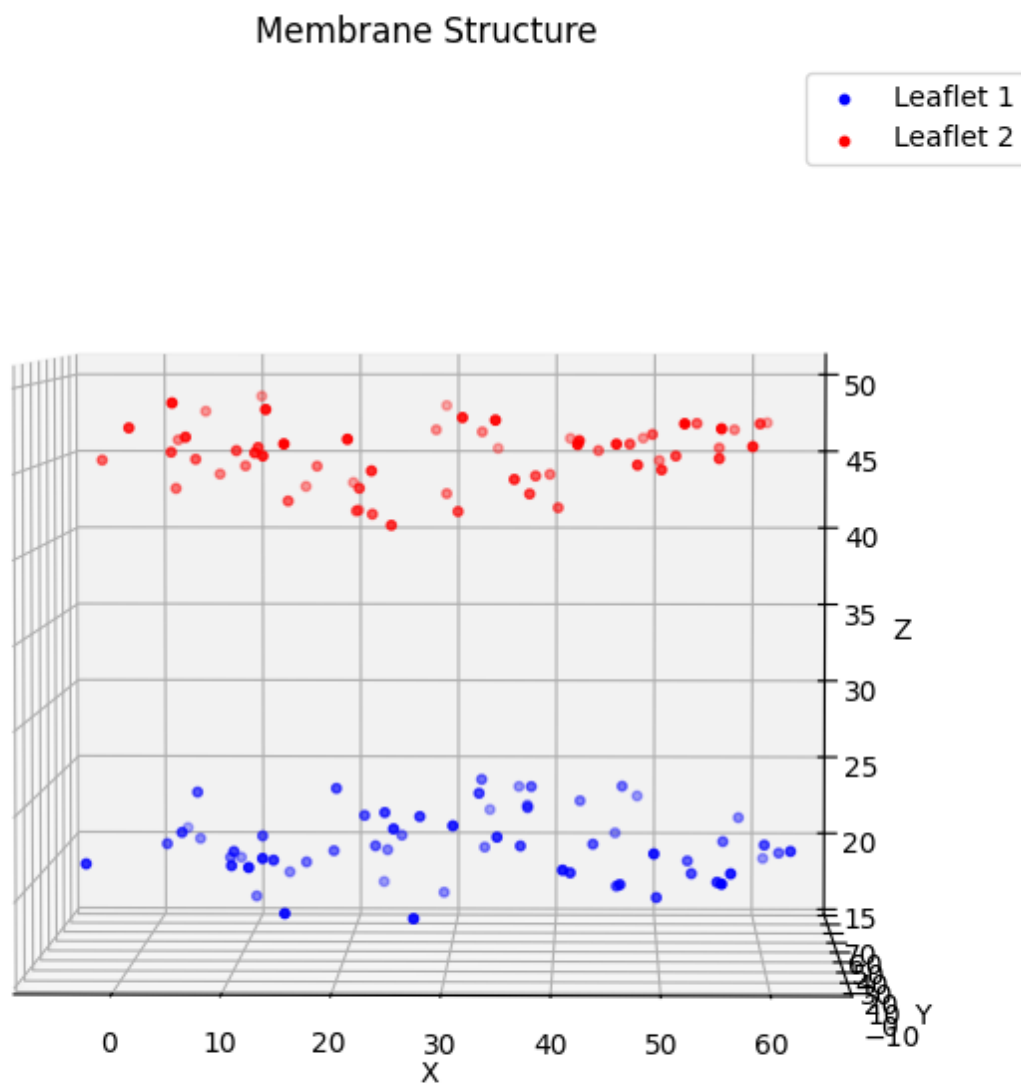


Figure 8. Résultat Centre de masse : bilayer\_peptide

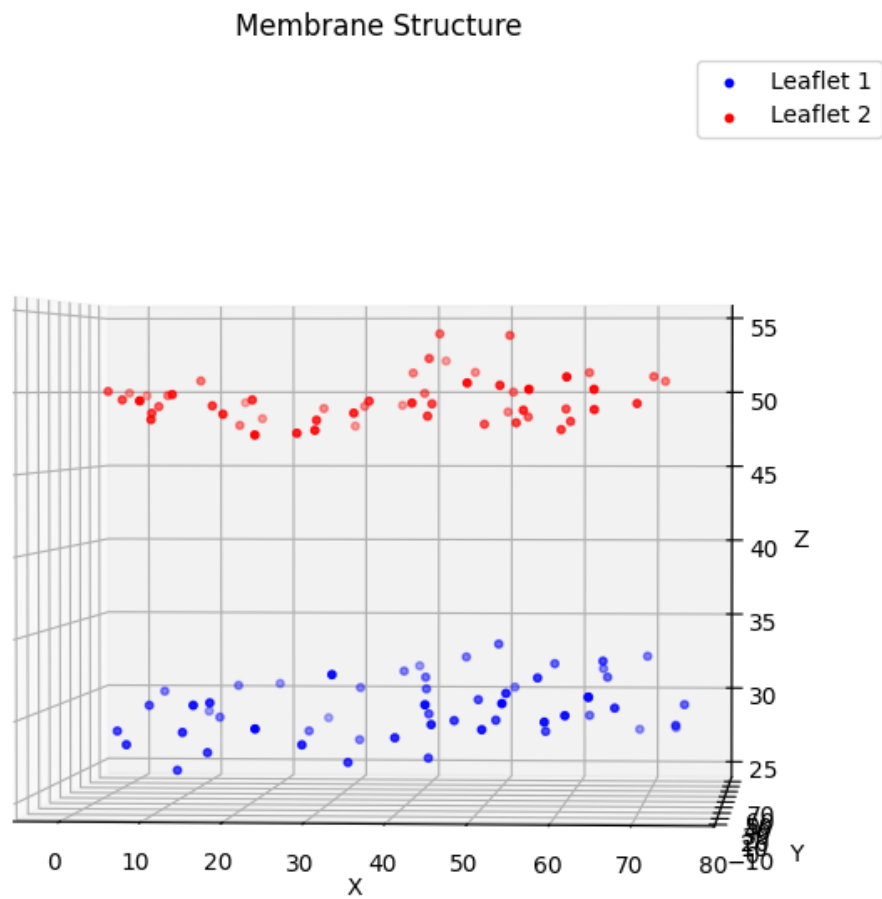


Figure 9. Résultat Centre de masse : bilayer\_prot

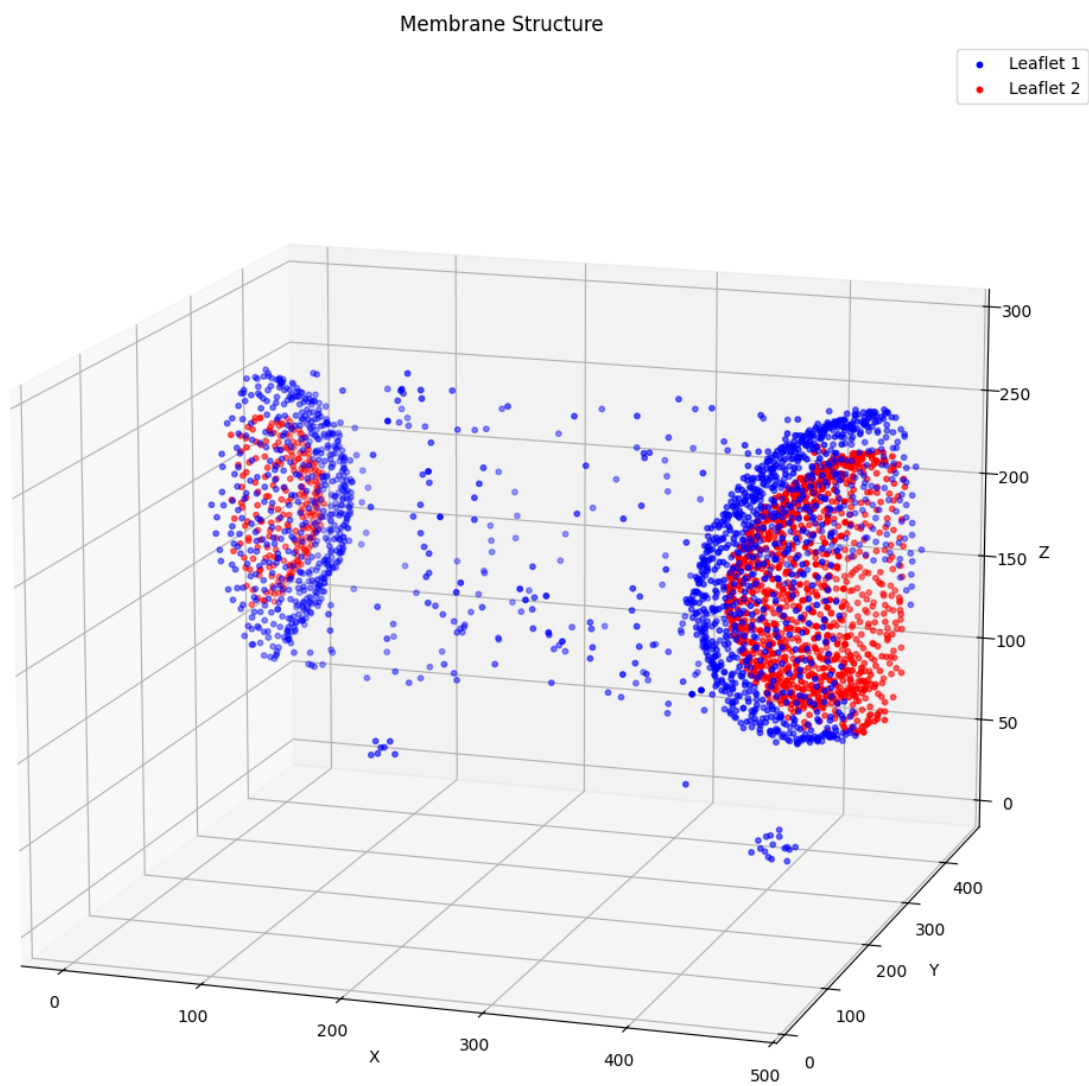


Figure 10. Résultat Centre de masse : dppc\_vesicle



## Raw Membrane Structure Before Leaflet Computation

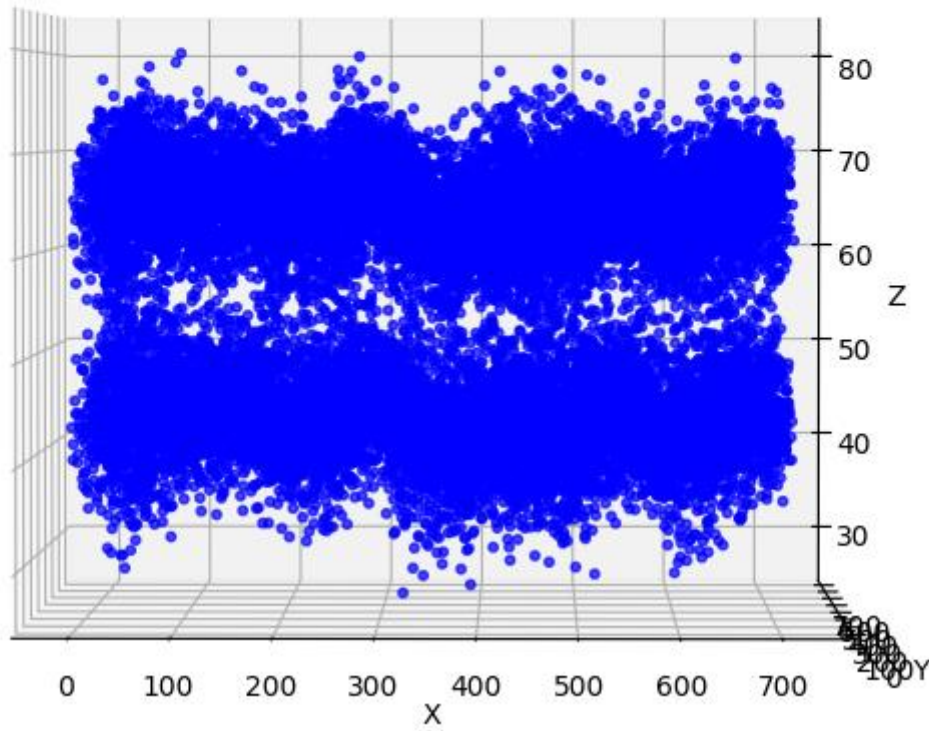


Figure 11: Résultat Centre de masse : *large\_plasma\_membrane*



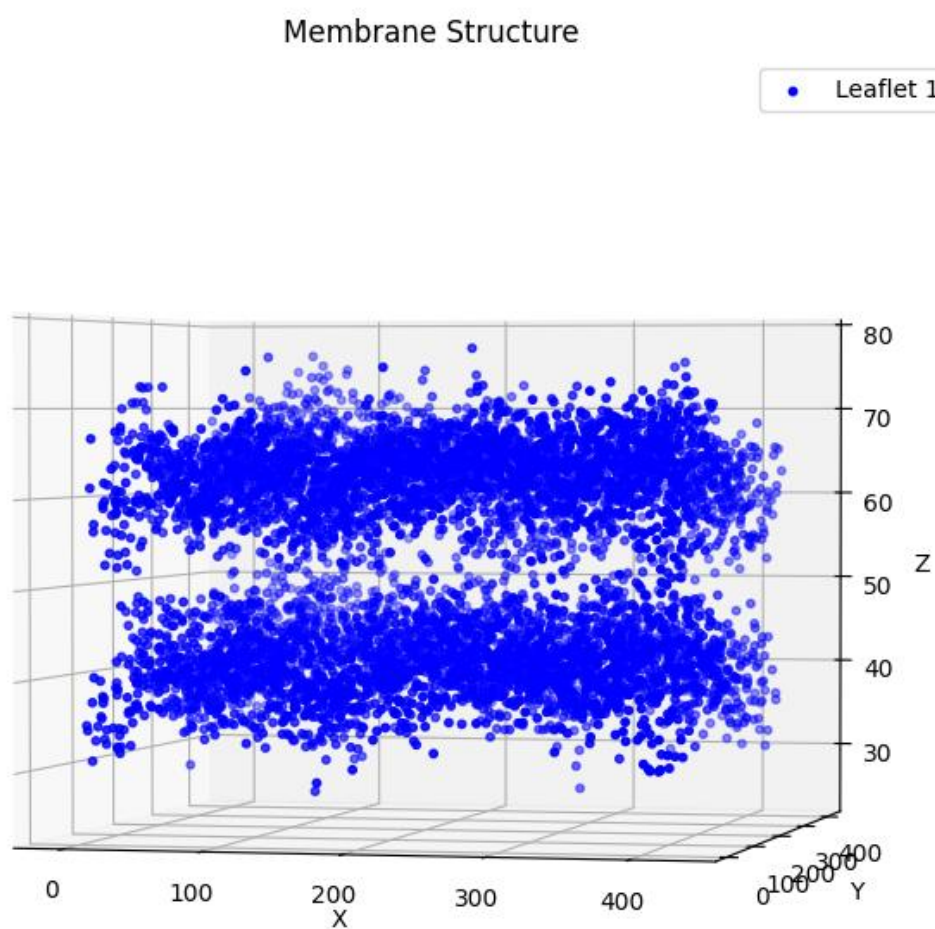


Figure 12. Résultat Centre de masse : *small\_plasma\_membrane*

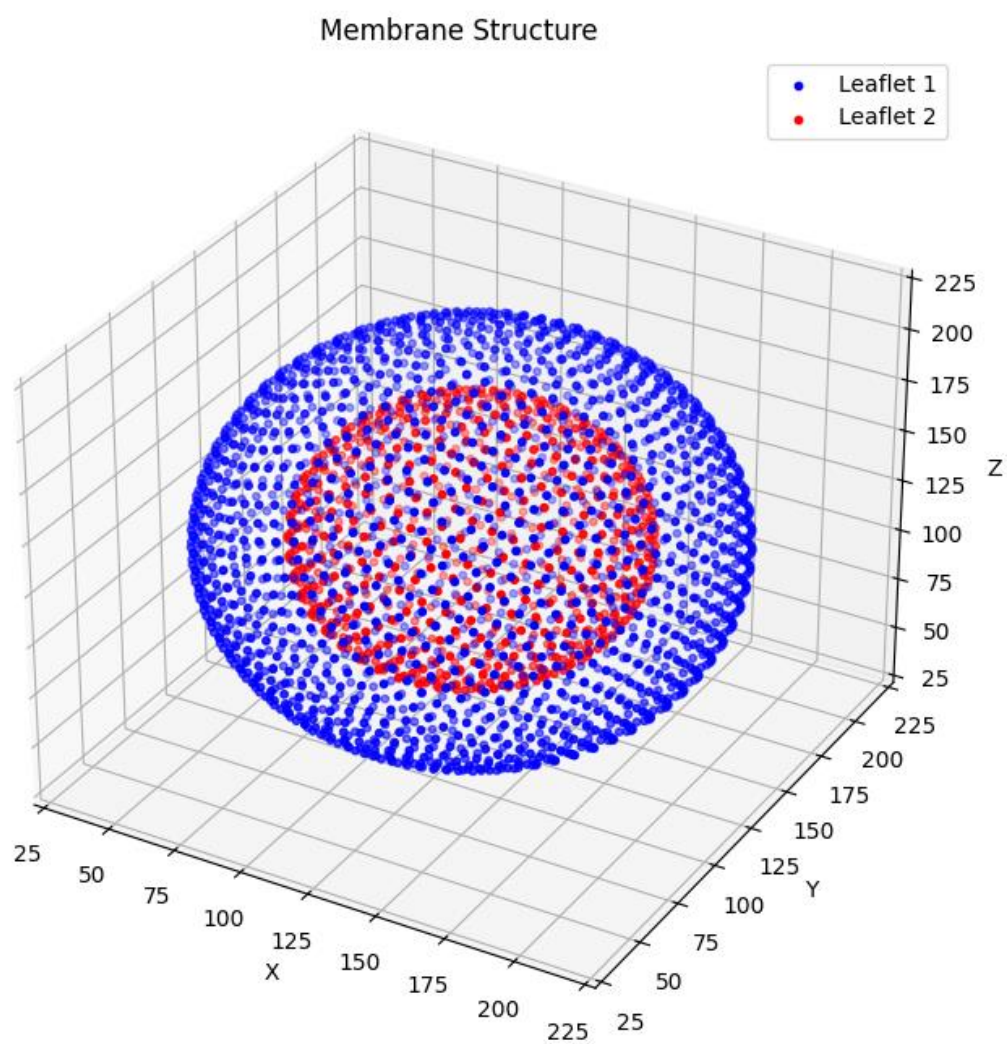


Figure 13. Résultat Centre de masse : model\_vesicle

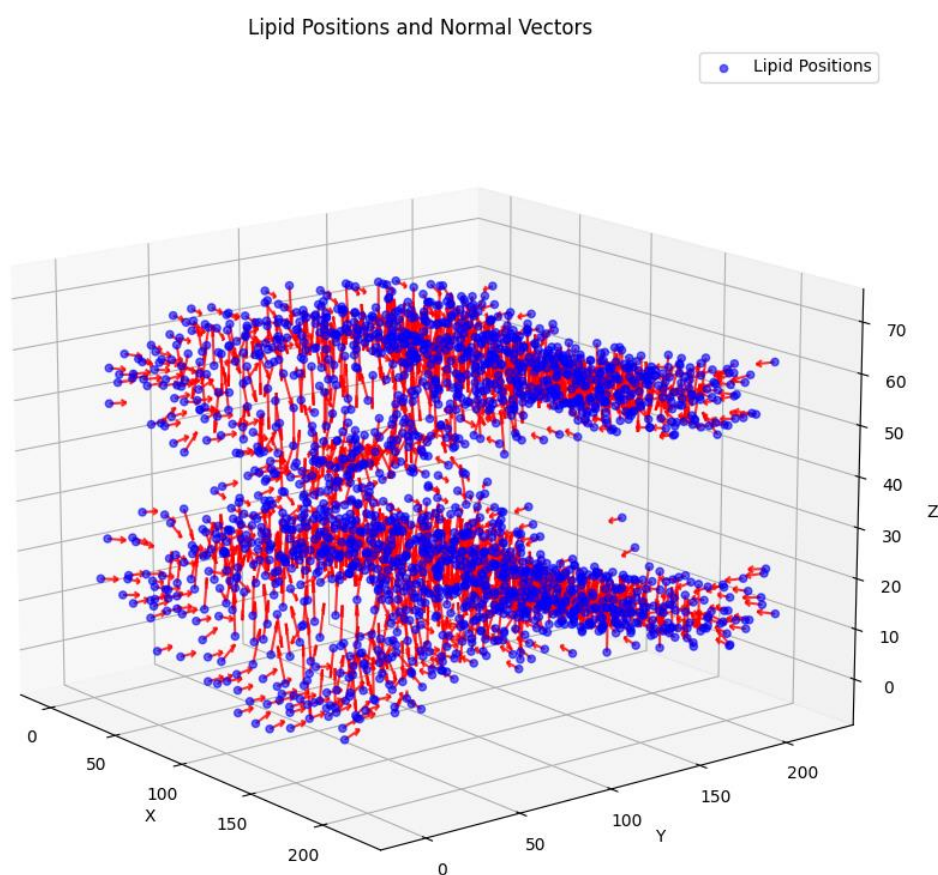


Figure 14. Résultat orientation vecteur normale : Bilayer Chol

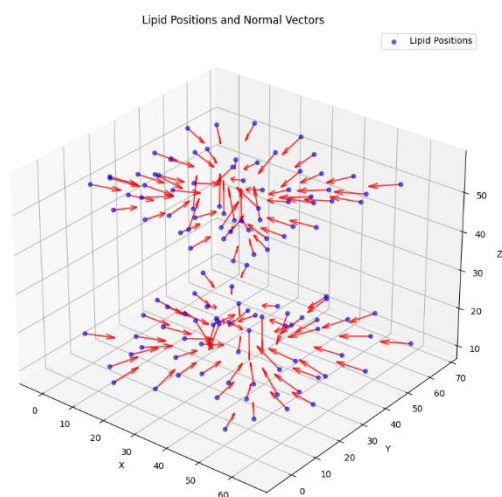


Figure 15. Résultat orientation vecteur normale : Bilayer Peptide

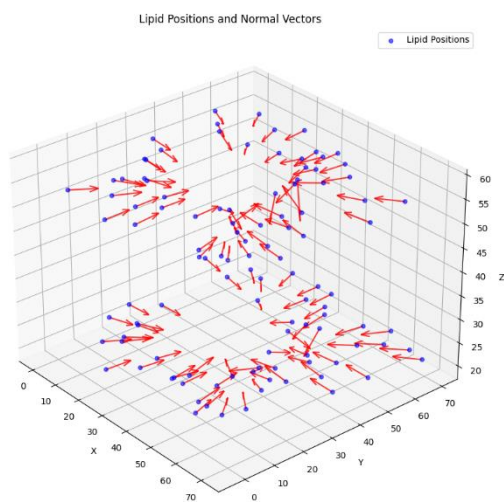


Figure 16. Résultat orientation vecteur normale : Bilayer Proteine

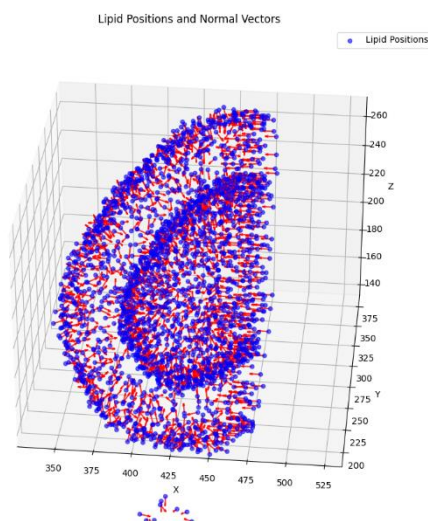


Figure 17. Résultat orientation vecteur normale : Zoom sur la partie droite de la membrane de DPPC Vesicule

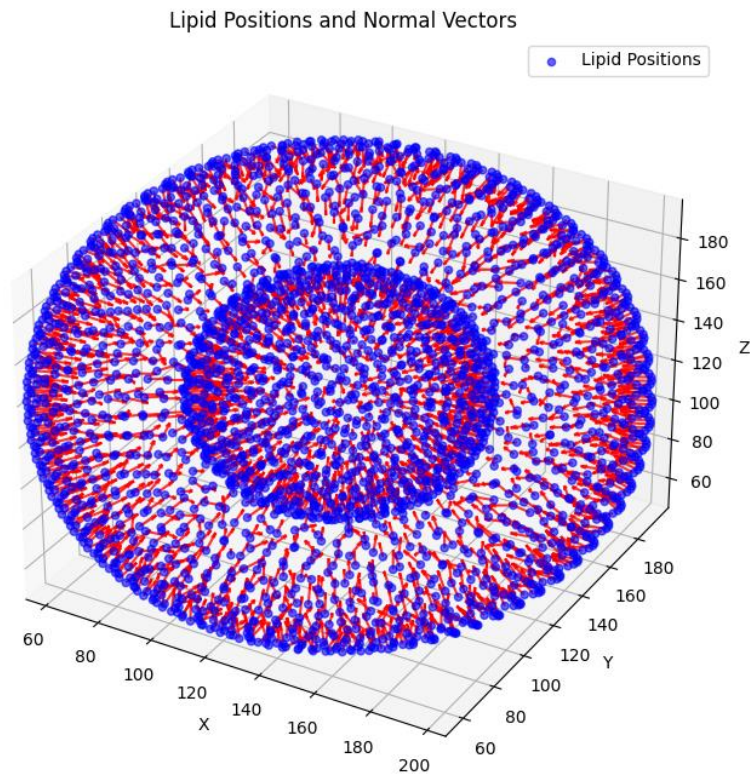


Figure 18. Résultat orientation vecteur normale : Vesicule Model



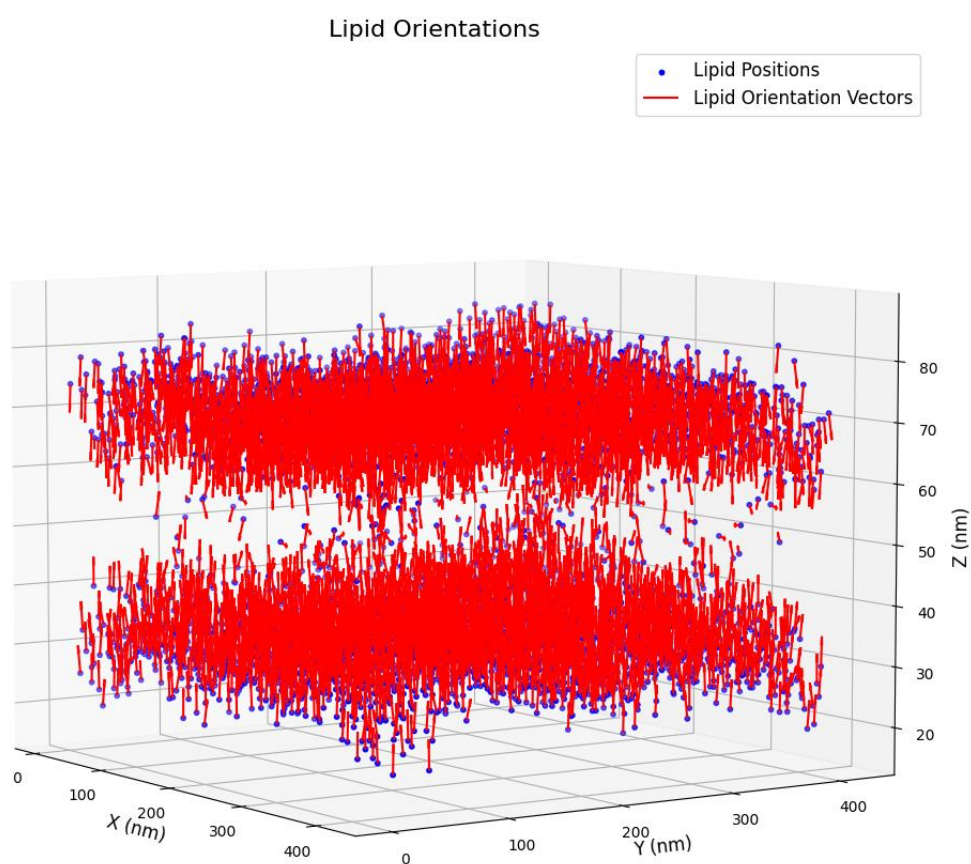


Figure 19. Résultat orientation sans vecteur normale local : Small Plasma Membrane

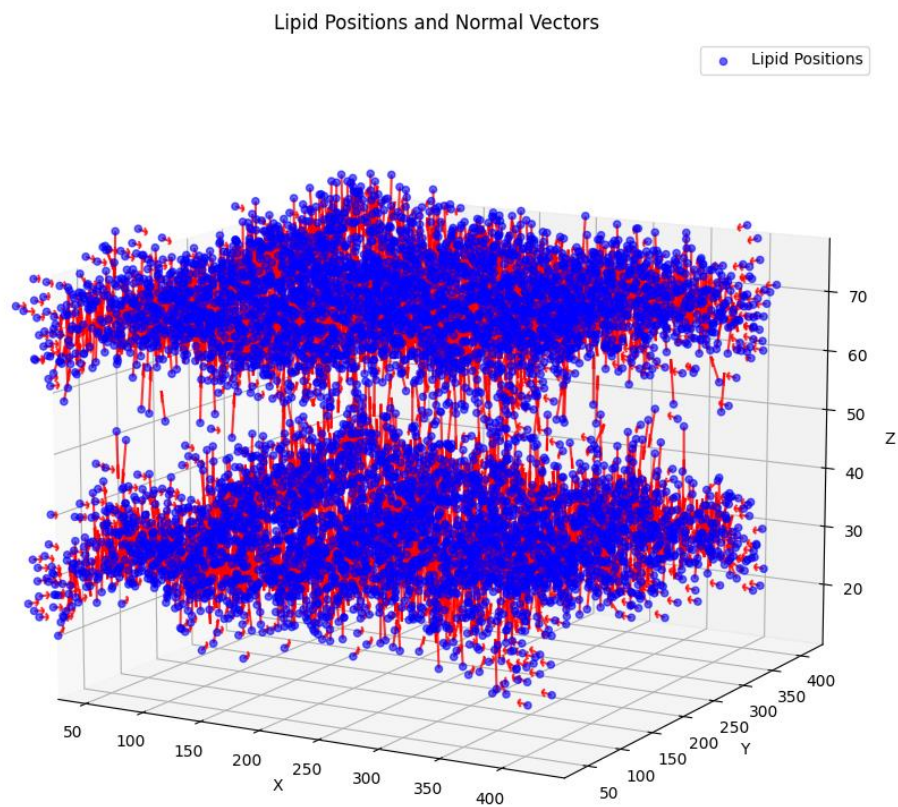


Figure 20. Résultat orientation avec vecteur normale local : Small Plasma Membrane



### Lipid Orientations

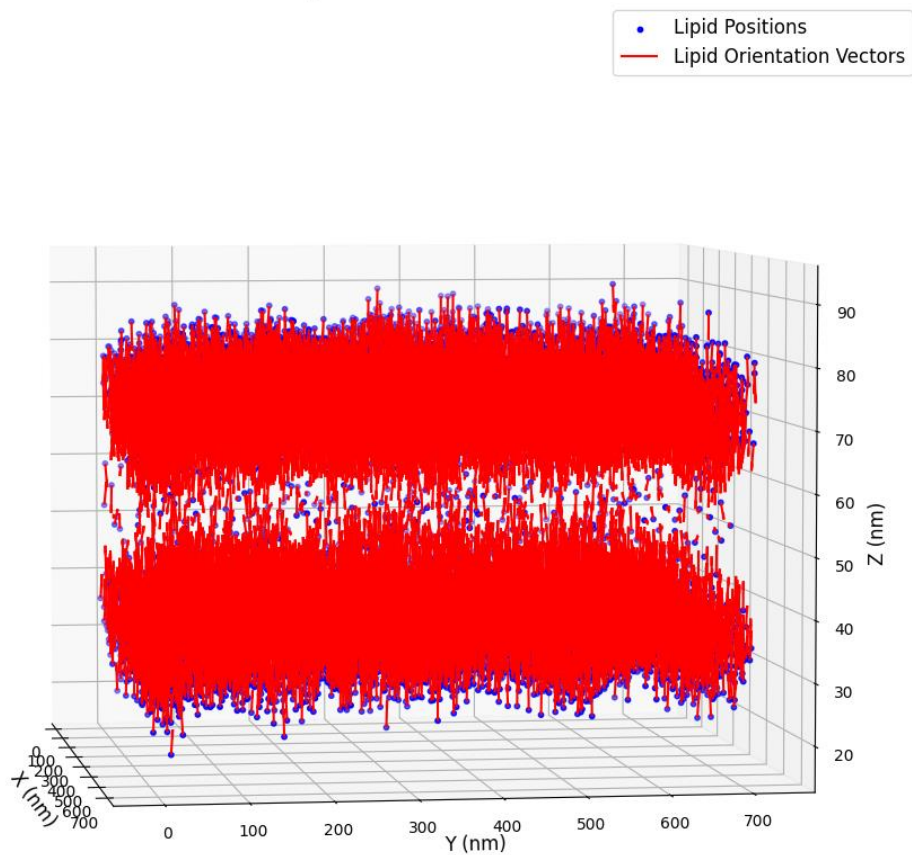


Figure 21. Résultat orientation sans vecteur normale local : Large Plasma Membrane