Gehao Hua, Helena Melcher, Paulène Riou
Projet 5BIM 2025/2026
Tutrice : Juliana Silva Bernardes - maître de conférences Sorbonne Université

Améliorer l'annotation fonctionnelle de génomes marins par l'intelligence artificielle

Données disponibles sur le drive :
https://drive.google.com/drive/folders/1S-w6QkmLNIy8hLdraUAW2BCiU1p0fblj?usp=drive_link

Dans le cadre de l'amélioration de l'annotation fonctionnelle de génomes de cyanobactéries, les méthodes actuelles peine à effectuer des prédictions fiables en raison de plusieurs limites. Afin d'explorer une nouvelle approche, il serait intéressant d'essayer d'appliquer une nouvelle méthode issu d'un article scientifique[1] récent dont les résultats obtenus (sur un autre organisme non modèle *Plasmodium*) sont très encourageants.
Dans ce rapport, nous présenterons en premier lieu le modèle PlasmoFP fondé sur le deep learning et les résultats obtenus dans cet article. Dans un second temps, nous développerons notre adaptation du modèle à nos données de cyanobactéries, les résultats et les perspectives.

Revue Bibliographique

Lien vers l'article : https://www.biorxiv.org/content/10.1101/2025.09.12.675843v1.full

Lien vers le github PlasmoFP_public : https://github.com/harshstava/PlasmoFP_public

Le génome de *Plasmodium* est séquencé depuis 2002. Traditionnellement, l'annotation fonctionnelle des protéines inconnues est effectuée de manière expérimentale. Cependant, cette méthode est peu pratique et n'est pas applicable pour toutes les protéines inconnues car elle nécessite beaucoup de ressources et de temps. C'est pourquoi, les techniques d'annotation fonctionnelle automatisées représentent une alternative indispensable pour permettre des avancées dans la recherche fondamentale sur le paludisme (dû aux parasites du genre *Plasmodium*).

PlasmoFP (pour Plasmodium Function predictor) est une approche AFP basée sur le deep learning. L'objectif de PlasmoFP est de prédire les différents termes GO de protéines caractéristiques des parasites *Plasmodium* pour 19 espèces différentes (organismes eucaryotes non modèles appartenant au groupe SAR). Pour cela, les modèles ont été entraînés sur les 3 sous-ensembles de termes GO (fonction moléculaire MF, procédé biologique BP et composant cellulaire CC dans Gene Ontology).

[1] Harsh R. Srivastava et al., PlasmoFP: leveraging deep learning to predict protein function of uncharacterized proteins across the malaria parasite genus bioRxiv 2025.09.12.675843; doi: https://doi.org/10.1101/2025.09.12.675843

La plupart des approches AFP (Automatic Function Prediction) reposent sur des méthodes de comparaison de similarité de séquences (comme BLAST) afin de prédire la fonction des protéines (grâce aux termes GO). Cette méthode montre rapidement ses limites : elle devient inefficace pour les protéines présentant une faible similarité avec les séquences déjà caractérisées. Ces protéines sont souvent uniques, ou bien ne présentent de similitude qu'avec des protéines faiblement annotées provenant d'autres parasites, voire avec des protéines très distantes appartenant à d'autres espèces, même si bien annotées. Et c'est ce qui est particulièrement retrouvé pour les protéines des espèces de *Plasmodium*. De plus, les protéines de *Plasmodium* peuvent contenir de longues régions intrinsèquement désordonnées, ce qui limite encore l'alignement avec d'autres protéines. C'est pourquoi, aujourd'hui encore, une grande partie des protéines de *Plasmodium* restent très peu caractérisées : 16% des protéines de *Plasmodium* n'ont aucune annotations fonctionnelle et 42% sont partiellement annotées.

PlasmoFP propose d'utiliser plus particulièrement la relation entre la structure et la fonction des protéines plutôt que la comparaison de séquences uniquement en se basant sur le fait que la structure est mieux conservée au cours de l'évolution. Leur modèle est également entraîné sur le supergroupe SAR (Stramenoliles, Alevrolates et Rhizarians) qui correspond à un groupe phylogénétique particulièrement pertinent pour l'étude de *Plasmodium* (car appartenant à la division Alveolata). Le jeu de données obtenu est donc très grand (autour de 700 000 séquences), ce qui permet d'entraîner le modèle avec une base de données suffisamment large. Ceci permet d'entraîner le modèle avec suffisamment de données.

La première partie de PlasmoFP consiste à répartir chaque séquence issue du jeu de données initial (SAR récupéré depuis Uniprot) selon les 3 sous-ensemble (MF, BP et CC) et à construire les jeux de données de chaque sous-ensemble pour l'entraînement, la validation et le test. Les jeux de données sont spécifiques au sous-ensemble auquel ils appartiennent, par exemple, les séquences des jeux de données pour le modèle MF ont été sélectionnées car elles sont annotées avec des termes GO du sous-ensemble MF. Afin d'éliminer les séquences redondantes, MMseqs2 est utilisé (90% de similarité). Les termes GO de chaque séquences ont été propagés vers le nœud racine correspondant de chaque sous-ensemble afin de récupérer les termes GO "ancêtres". Pour réaliser cette étape, PlasmoFP utilise le service QuickGO afin de trouver tous les ancêtres grâce à la relation "is_a". MMseq2 est à nouveau utilisé pour effectuer un clustering à 30% de similarité afin de s'assurer que les séquences entre les jeux de données d'entraînement, de validation et des tests ne soient pas trop semblables.

La deuxième étape consiste à la création des embeddings TM-Vec pour chaque séquence protéique. Les séquences sont encodées via le modèle TM-Vec 'CATH' qui permet la génération d'un embedding sous forme de vecteur Z de longueur 512 (dimension fixe et compacte de (1,512)) pour chaque séquence. Le modèle TM-Vec se base sur des modèles d'apprentissage profonds tels que ProtT5, qui a montré un fort potentiel pour encoder des protéines — qu'elles soient décrites par leur séquence ou leur structure tridimensionnelle. TM-Vec est entraîné en prédisant la similarité structurale de deux embeddings de protéines de ProtT5. Les représentations internes dans le modèle entrainé sont utilisées comme embeddings. Les résultats sur *Plasmodium* ont montré une augmentation significative de la

performance du modèle PlasmoFP avec l'utilisation de TM-Vec comparé aux autres modèles essayés.

La troisième étape consiste à entraîner les différents modèles correspondant à chaque sous-ontologie. Il repose sur un réseau multicouche. Plusieurs paramètres sont testés afin de trouver le modèle le plus performant : le nombre de couches, le learning rate et le nombre d'epoch. Chaque modèle est évalué sur sur les jeux de données d'évaluation correspondant à chaque sous-ontologie en utilisant les métriques Fmax et Smin de CAFA (le modèle sélectionné étant celui avec le meilleur set de ces deux métriques).

Le modèle permet d'attribuer une probabilité à chaque terme GO pour chaque séquence protéique (probabilité pour chaque paire protéine-terme GO) ainsi qu'un FDR (false discovery rate) observé. Un seuil FDR attendu (eFDR) est spécifié par l'utilisateur (20% dans leur cas) et permet de sélectionner les termes GO attribués aux protéines partiellement annotées ou inconnues selon la probabilité ajustée calculée. Ce qui signifie qu'il n'y a pas de seuil unique de FDR pour tous les termes GO mais qu'il est spécifique à chaque terme.

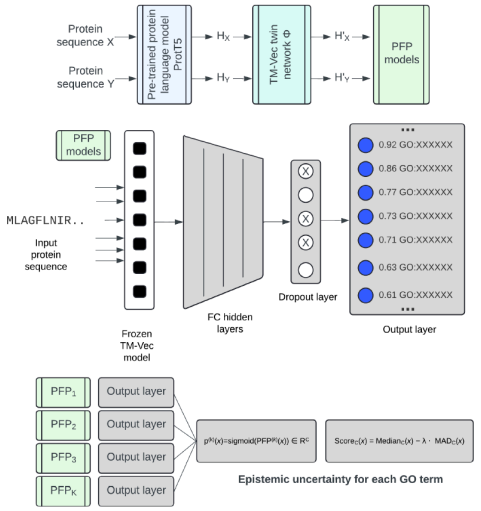


Fig 1 | PlasmoFP models train on structure representations and utilize deep ensembles (A) Schematic representation of the workflow for training PlasmoFP models using protein language model embeddings. Protein sequences (e.g., X and Y) are processed through the pre-trained ProtT5 language model to generate embeddings (H_X and H_Y). These embeddings are then fine-tuned through the TM-Vec twin network (Φ) to produce sequence-specific representations (H'_X and H'_Y), which serve as inputs to the PlasmoFP models during model training and inference. (B) PlasmoFP models predict GO term probabilities from the processed sequence embeddings (H') via fully connected (FC) layers. (C) Each PlasmoFP model for each subontology is configured as a deep ensemble trained on different *k* folds of the training dataset. We calculate per-term epistemic uncertainty and create a conservative uncertainty adjusted probability for each input-GO term pair.

Résultats de PlasmoFP sur *Plasmodium* :
Résultats de prédiction pour toutes les espèces disponible sur le github PlasmoFP_Explorer : https://github.com/harshstava/PlasmoFP_Explorer

D'après leurs résultats, le modèle proposé par PlasmoFP surpasse les autres méthodes existantes (ProeInfer, DeepGo) sur les 3 sous-ontologies pour *Plasmodium*. De plus, les performances de PlasmoFP ne baissent que très faiblement pour les séquences avec un niveau de désordre intrinsèque élevé. Cela suggère que les modèles apprennent bien des caractéristiques structure-function pertinentes ce qui permet de prédire leur fonction même dans le cas où les méthodes traditionnelles basées uniquement sur les séquences échouent.
Dans l'ensemble, l'annotation fonctionnelle des différentes espèces du genre *Plasmodium* a été améliorée. PlasmoFP a donc été utilisé pour prédire les termes GO pour les protéines étiquetées "proteins of unknown function" (PUFs), c'est-à-dire complètement non-annotées, et les protéines partiellement annotées. La part de protéines annotées sur les trois sous-ontologies a été augmentée en moyenne de 35,3% et la part de protéines non-annotées

a été réduite de 25,3% en moyenne par espèces. PlasmoFP a particulièrement permis d'améliorer les annotations fonctionnelles des espèces de *Plasmodium* très peu annotées initialement, les rapprochant du niveau d'annotation des espèces les plus annotées. Les gains pour les espèces les plus annotées initialement ont les gains absolus les plus faibles. Globalement, la proportion de protéines complètement annotées est passée de 7-42% à 36-68% entre les espèces. De même, la proportion de protéines non-annotées a été bien réduite puisqu'elle est passée de 15-59% à 3-28%.

De plus, les annotations prédites font sens biologiquement d'après l'évaluation des nouveaux termes GO en se basant sur les annotations existantes. Cela permet de confirmer que les termes GO attribués par PlasmoFP ont une forte cohérence biologique.

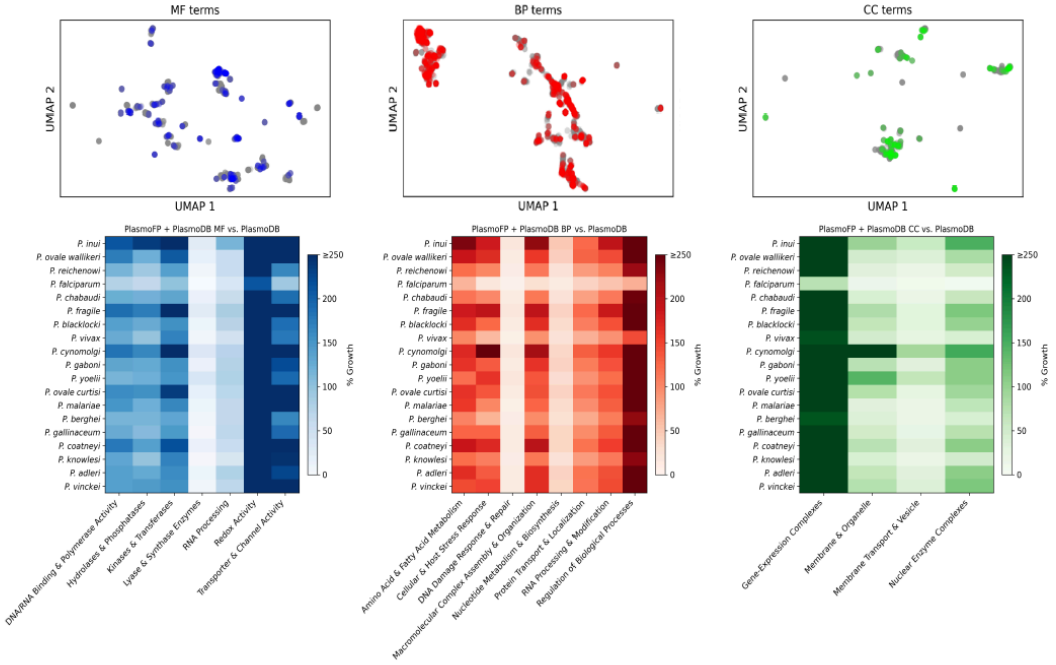


Fig. 4 | PlasmoFP predicted GO terms co-cluster with existing annotations and expand functional categories across subontologies (A) UMAP projections of Resnik semantic-distance embeddings for GO terms in each subontology (MF: blue, BP: red, CC: green). Gray points are existing annotations; colored points terms predicted by PlasmoFP not found in the existing PlasmoDB annotation across all 19 *Plasmodium* species (B) Heatmaps showing the increase in annotated proteins per functional cluster in the PlasmoFP + PlasmoDB set verses the PlasmoDB annotati

PlasmoFP est donc un modèle très performant mais il n'existe pas encore de méthode permettant d'annoter de bout en bout les protéines non caractériser avec une très faible homologie de séquence avec les protéines bien caractérisées au sein d'un sous-ensemble d'espèces défini sur le plan phylogénétique. *Plasmodium* étant un ensemble d'espèces non modèles, les approches telles que DeepGOPlus et ProteInfer, qui se basent sur un ensemble d'organismes modèles, ont des résultats avec des écarts plus importants que PlasmoFP en raison des homologies plus éloignées. PlasmoFP est une approche qui permet à la fois de résoudre les problèmes d'homologie éloignée et la représentation d'organismes non modèles pendant l'entraînement.

L'approche de construction d'un modèle de prédiction fonctionnelle spécifique à une espèce comme PlasmoFP peut être généralisée à d'autres protéomes non modèles.

Application de la méthode PlasmoFP à nos données

Notre problématique

Les organismes marins comme les cyanobactéries sont des micro-organismes très abondants sur Terre et essentiels à la survie de la biodiversité complexe du milieu marin. Afin de mieux appréhender les différents enjeux qui s'y jouent, il est important de comprendre au mieux le rôle des protéines qui composent le protéome de ces micro-organismes. Cependant, bien que de nombreux génomes des cyanobactéries ou encore des algues brunes soient référencés, l'annotation fonctionnelle de ces génomes reste encore largement incomplète.

Les génomes des organismes marins sont souvent sous-annotés en raison de la grande diversité phylogénétique et de la spécificité écologique des espèces marines. Beaucoup d'entre elles appartiennent à des clades peu étudiés, dont les gènes ne présentent aucune homologie significative avec les séquences déjà connues. De plus, les conditions extrêmes de l'environnement marin (comme haute pression, salinité, température extrême) permettent l'apparition de protéines uniques, qui sont difficiles à relier aux fonctions de référence préexistantes.

L'annotation de génomes marins est donc essentielle pour comprendre le rôle fonctionnel de la biodiversité marine, révéler de nouvelles enzymes ou voies métaboliques, et mieux modéliser les cycles biogéochimiques globaux (carbone, azote, soufre…). Ces annotations permettent également d'améliorer les bases de données biologiques, ce qui facilitera l'identification automatique de nouveaux gènes issus d'autres génomes marins.
C'est pourquoi, il est nécessaire de mettre en place de nouveaux outils plus performants afin d'améliorer cette annotation.

Adaptation à notre situation

Le modèle de PlasmoFP, adapté aux cyanobactéries, peut être une approche qui permettrait d'améliorer l'annotation fonctionnelle déjà existante. Comme pour *Plasmodium*, les cyanobactéries sont des organismes non modèles, ce qui rend les méthodes d'annotation fonctionnelle "classiques" peu performantes. De la même manière que l'approche de PlasmoFP, nous avons entraîné le modèle de sorte à ce qu'il convienne au mieux aux types de séquences protéiques spécifiques des cyanobactéries. C'est pourquoi, nous avons choisi d'entraîner le modèle avec une base de données issues d'Uniprot reprenant la plupart des séquences protéiques des bactéries. Le choix du domaine des bactéries s'explique par le fait que les cyanobactéries appartiennent à ce domaine (comme *Plasmodium* appartient au groupe SAR).

Comme dans PlasmoFP, nous avons utilisé des données téléchargées de uniprot ([Entry_name, Organism, Sequence, Gene Ontology (MF), Gene Ontology (CP), Gene Ontology (CC), Pfam, Interpro (taxonomy_id:2) AND ((existence:3) OR (existence:2) OR (existence:1))

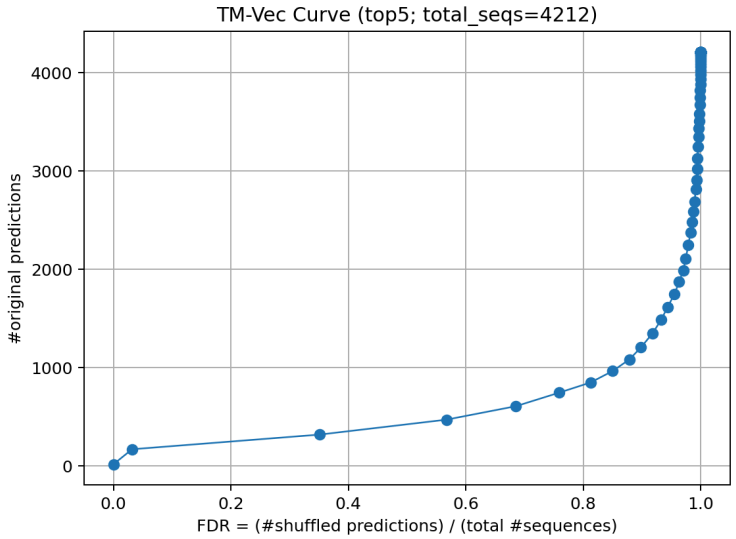
AND (length:[* TO 1200])]) de la clade phylogénique de nos espèces d'intérêt. Les filtres par score d"existence correspondent à de l'évidence expérimentale au niveau des protéines (1) ou des transcript (2), ou des protéines inférées par l'homologie (3). Pour rassembler assez de données, nous avons utilisé les séquences venant de bactéries. Cela nous a laissé avec un dataset bien plus petit que celui de SAR (environ 330 000 au lieu 800 000), mais inclure des espèces eucaryotes, ou avec des annotations de moindre qualité, ne nous semblait pas être très pertinent pour l'entraînement d'un modèle à appliquer aux cyanobactéries.

Nous avons un jeu de données de séquences de protéines venant de cyanobactéries livré par la cliente. Notre but est d' annoter ces séquences avec des termes GO. Pour tester la fiabilité des prédictions, nous avons un autre jeu de données, qui correspond au premier avec l'ordre des acides aminés mélangés (données shuffled). Idéalement, notre modèle devrait produire peu de prédictions pour ces séquences.

Modèle simple avec TM-Vec

Le modèle le plus simpliste est un modèle d'homologie. Il prédit quelle séquence d'entraînement est la plus similaire à la séquence à prédire, et assigne le termes GO de la première séquence à la deuxième. Pour ceci, nous avons utilisé les embeddings de TM-Vec, et la fonctionnalité de TM-Vec search, qui obtient les embeddings les plus similaires dans une base de données. Une prédiction ici correspond donc à la séquence la plus similaire, et son score est la similarité des deux séquences.

En utilisant un script Python *fdr_plot_new.py*, le graphique suivant est obtenu :



Cette courbe est tracée à partir des résultats de recherche TM-Vec en considérant, pour chaque séquence query, ses cinq voisins les plus proches. Pour un seuil donné de TM-score, une séquence est comptée comme un hit, si au moins une correspondance dans la base originale dépasse ce seuil. La même procédure est appliquée à une base ce séquences shuffled afin d'estimer le nombre de faux positifs. Le taux de fausses découvertes (FDR) est défini comme le rapport entre le nombre de hits obtenus sur la base shuffled et le nombre

total de séquences query. Le nombre de prédictions originales est représenté en fonction de ce FDR.

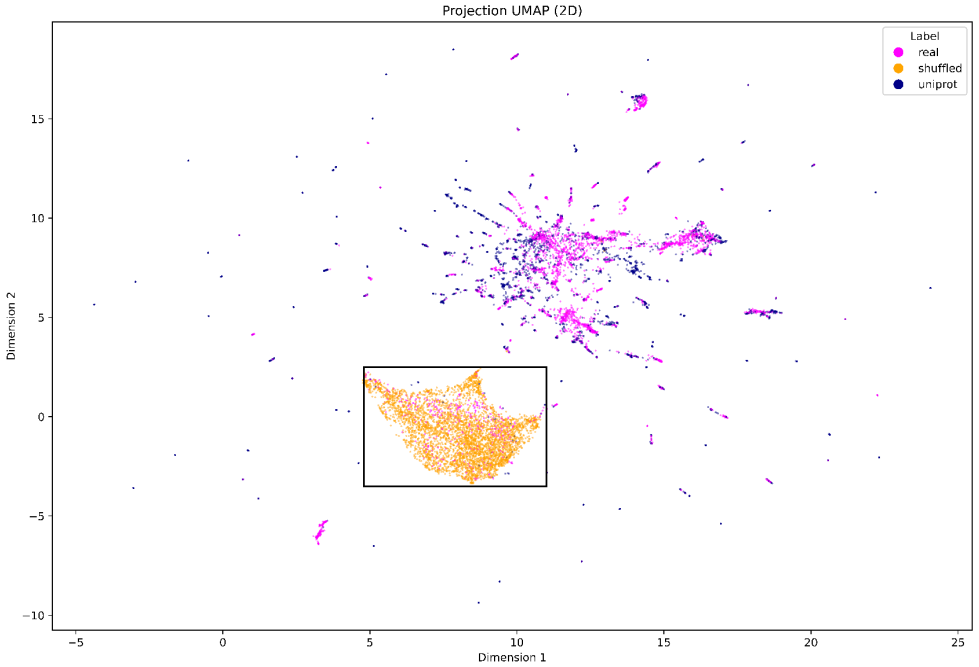
La courbe montre que quand le FDR augmente, le nombre de prédictions dans la base originale est strictement croissant. Cette croissance monotone peut montrer l'assouplissement progressif du seuil de similarité. On observe un nombre substantiel de prédictions à faible FDR, ce qui indique que TM-Vec capture des similarités structurales significatives au-delà du hasard. Cette courbe est une référence méthodologique pour les analyses fonctionnelles ultérieures, qui reposent sur la même définition du FDR.

Cependant, cette courbe met en évidence une limite importante de TM-Vec. On observe que le nombre de prédictions originales augmente de manière significative uniquement lorsque le FDR est très élevé, c'est-à-dire dans une région où il y a beaucoup de faux positifs. Le modèle simple avec TM-Vec n'a donc pas trop d'intérêt pratique pour des analyses nécessitant un haut niveau de confiance.

Représentation des embeddings TM-Vec par un UMAP

La visualisation 2D des embeddings des séquences protéiques sur un UMAP nous permet de comparer la répartition des embeddings en fonctions types de séquences :

- Les données réelles des cyanobactéries (issues de Cyanorak)
- Les données shuffled des séquences de cyanobactéries, c'est-à-dire que les séquences sont les séquences des données réelles mais complètement randomisées.
- 10% des données Uniprot (les séquences protéiques de bactéries)



Il est important que les embeddings générés par TM-Vec soient différentiables. Cela se traduit sur le graphique UMAP par une séparation entre les catégories sur le plan 2D.

Nous pouvons observer, sur notre graphique, que les données shuffled sont bien regroupées dans une zone spécifique. La zone, délimitée par le rectangle noir sur la figure, permet de regrouper la quasi totalité des embeddings des séquences shuffled mappées sur le UMAP. Ceci veut dire que les embeddings de TM-Vec de séquences sans sens sont tous similaires. Globalement, les embeddings des deux autres jeux de données (réels et Uniprot) ne sont pas mappés dans la même zone que ceux des séquences shuffled. Cependant, il existe des exceptions. Entre autres, nous retrouvons environ 370 protéines issues des données réelles qui sont regroupées dans la même zone que celles du shuffled sur plus de 4000. De même, sur 30 000 séquences de bactéries mappées (soit 10% de notre jeu de données), environ 4500 sont retrouvées dans la zone identifiée caractéristique des séquences shuffled.

Appropriation du code de PlasmoFP

Nous avons adapté le code de base de PlasmoFP pour entraîner des modèles nous-mêmes sur nos données.
PlasmoFP entraîne un modèle par sous-ensemble, donc trois modèles différents. Pour diminuer la complexité de notre tâche et par manque de temps, nous avons décidé de nous concentrer sur l'une des trois. Après discussion avec la cliente, nous avons choisi de nous concentrer sur le sous-ensemble Molecular Function.

Gestion de donnés

La première étape est de filtrer les séquences pour obtenir seulement celles qui ont des termes GO de l'ontologie Molecular Function (301 300). Après ça, mmseqs2 est utilisé pour clusteriser les séquences avec une similarité supérieure à 90 %. Pour chaque cluster, la séquence avec le plus de termes GO est choisie (150 860 séquences).

Comme la base de données de uniprot ne contient que les termes GO les plus spécifiques, il faut propager les termes vers leur parents. Pour ceci, le code de PlasmoFP utilise l'API de QuickGO. Plus tard, ils utilisent le fichier go-basic.obo téléchargé à partir du Gene Ontology Consortium pour calculer certaines statistiques, ce qui ne donne pas toujours des résultats équivalents. Pour cette raison nous avons changé le code pour aussi employer le fichier .obo. Nous avons aussi remarqué que la colonne de termes GO utilisés pour l'entraînement des modèles ne contient que les ancêtres des termes GO donnés par uniprot, mais non pas les termes eux-mêmes. Puisque ceci n'était pas mentionné dans l'article comme choix délibéré, nous avons décidé de changer le code à nouveau afin d'inclure tous les termes GO. Pour simplifier le traitement des termes GO nous n'avons pas remplacé les termes GO obsolètes.

MMseqs est utilisé une deuxième fois avec une similarité de 30. L'output sert à répartir les séquences de protéine en entraînement, validation, et test. 10 % des clusters sont assignés au jeu de données de test (test), 8 % à celui de validation (val), et 72 % pour le jeu données d'entraînement (train). Les termes GO avec moins de 50 apparences dans les séquences d'entraînement sons filtrés. Cela enlève 2947 termes GO, pour un résultat final de 1098 termes GO. Les embeddings TM-Vec sont calculés. Finalement, la statistique de Information

Accretion est calculée pour chaque terme GO. Elle représente le log négatif du nombre de protéines avec le terme GO, divisé par le nombre de protéines avec tous les parents du terme GO. En testant cette fonction avec notre fichier .obo, nous avons remarqué que le code de PlasmoFP renverse les parents et les enfants de termes GO dans nos essais. Pour résoudre ce problème, nous avons renversé la direction du graphe.

Entraînement du modèle

Nous avons copié les architectures et les hyperparamètres de PlasmoFP. Plusieure configurations sont testées pour trouver la meilleure. Le vecteur d'input est l'embedding de TM-Vec avec une longueur de 512. Notre jeu de données contient 1098 termes GO différents représentés par un encodage one-hot, comparé au 744 termes GO de fonction de PlasmoFP. Les architectures testés utilisent un dropout de 0.4 pour l'avant-dernière couche, le Leaky ReLU comme fonction de non-linéarité, et le Binary Cross Entropy loss.

Trois différents hyperparamètres sont testés: la Learning Rate (0,01 ; 0,001 ; 0,0001 ; 0,00001), l'architecture de Hidden Layers ([256] ; [256, 128] ; [256, 128, 64] ; [256, 128, 62, 32]) et le nombre d'epochs pour l'entraînement (10 ; 15 ; 20 ; 30 ; 40).

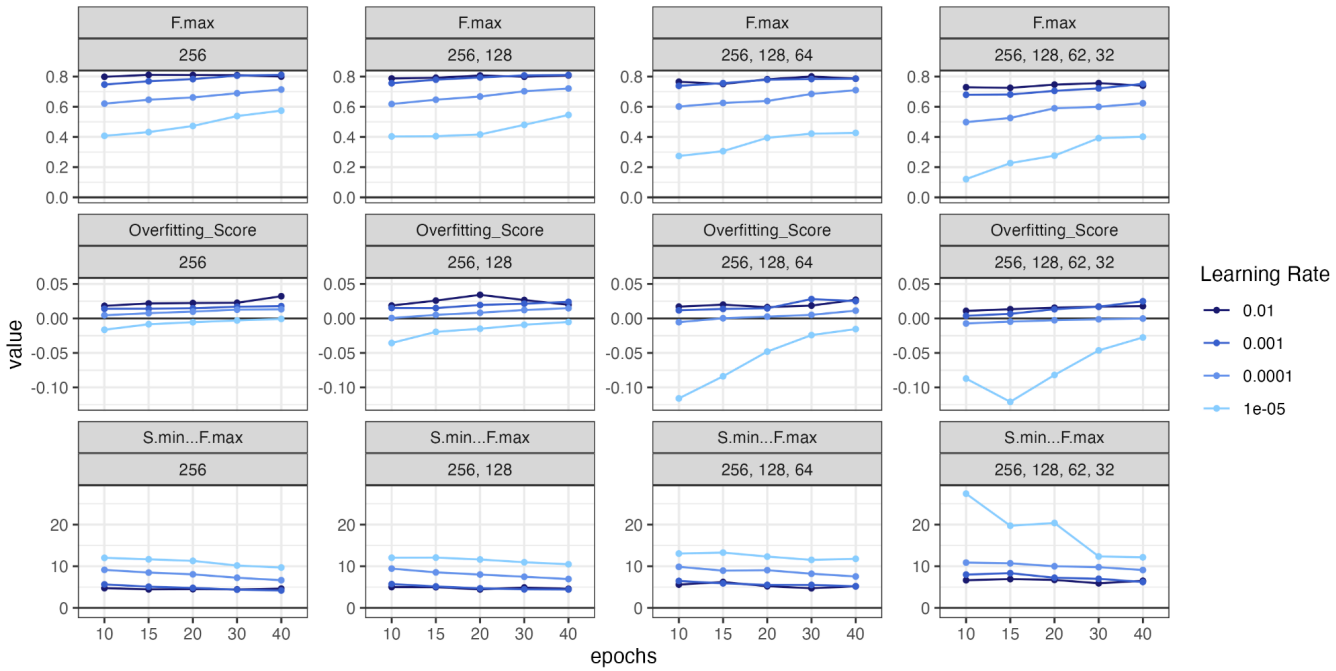
Choix du modèle

Afin de choisir le modèle le plus performant, nous pouvons nous baser sur 3 métriques différentes (2 métiques CAFA, Fmax et Smin et l'overfitting) calculées d'après les résultats de chaque modèle sur le jeu de validation correspondant au sous-ensemble MF :

- La métrique d'overfitting est la différence entre le loss de validation et le loss de training. Il permet de s'assurer que le modèle ne sur-apprend pas sur les données d'entraînement.
- La métrique Fmax (entre 0 et 1) doit être maximisée. Elle représente le meilleur score F1 (ratio entre précision et recall).
- La métrique Smin at Fmax doit être le plus bas possible. Smin mesure la distance sémantique (représenté par la Information Accretion) entre prédictions et réalités. Smin at Fmax est cette statistique au seuil avec le score F1 le plus haut.

D'après les résultats (voir la figure ci-dessous) que nous avons obtenus pour les différents modèles testés, nous pouvons faire plusieurs choix différents. Nous pouvons observer que, globalement, lorsque le learning rate diminue, les résultats dégringolent. D'autre part, les résultats ne semblent pas s'améliorer lorsque le nombre de couches dans la structure augmente. Finalement, nous avons choisi de garder le modèle avec une architecture [256], pour un learning rate de 0,001 et avec 30 epochs.

Nous avons observé des overfitting-score négatifs. Cela indique que le modèle produit de meilleurs résultats sur les données de validation, ce qui devrait être impossible. Comme ce résultat n'apparaît qu' avec des learning rates très petites, et diminue avec le nombre d'epochs d'entraînement, notre hypothèse est que ce résultat est dû à la configuration

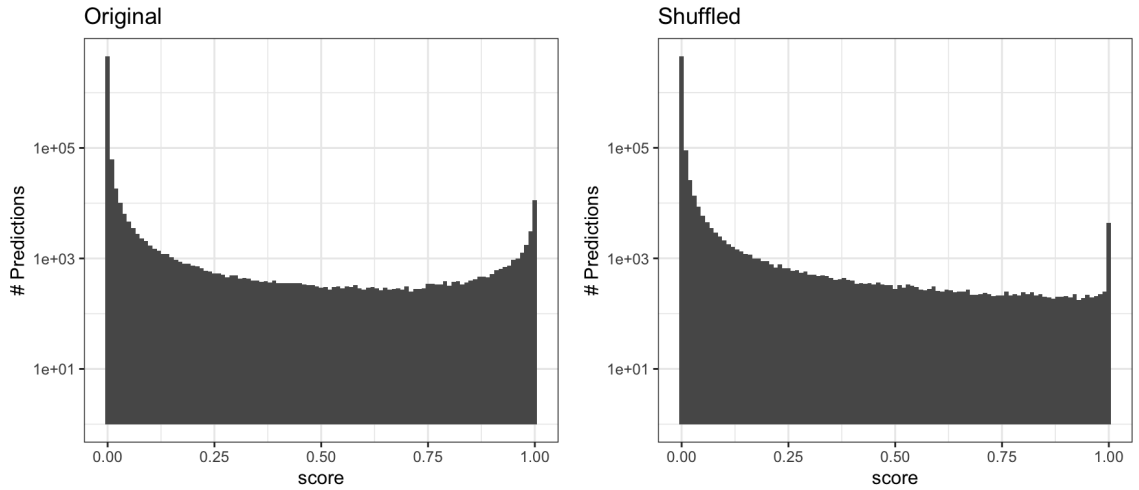


initiale. Le score de overfitting est aussi négatif chez les résultats de PlasmoFP pendant les premières epochs avec une learning rate en dessous de 0.001.

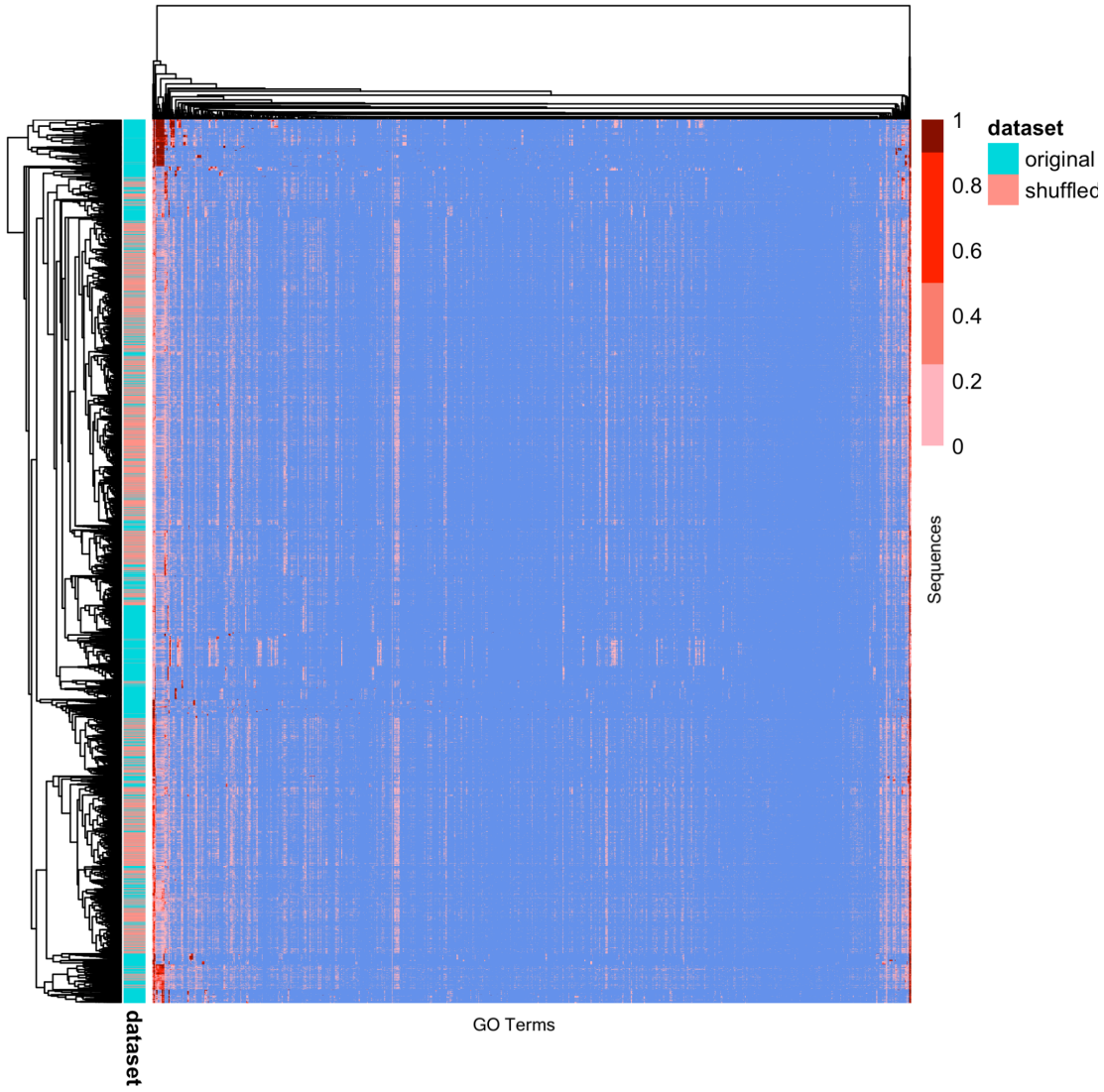
Pendant l'étape de tuning d'architecture, PlasmoFP essaie d'autres types de fonctions loss, comme un loss ajusté par la fréquence des termes GO, et les fonctions de loss Focal, Jaccard, Hinge, et Asymmetric. Comme dans leur résultats, ces alternatives n'améliorent pas les résultats.

Prédiction et évaluation

Pour tester les résultats de notre modèle, nous avons créé des prédictions des deux jeux de données de cyanobactéries (original et shuffled) avec le modèle choisi.



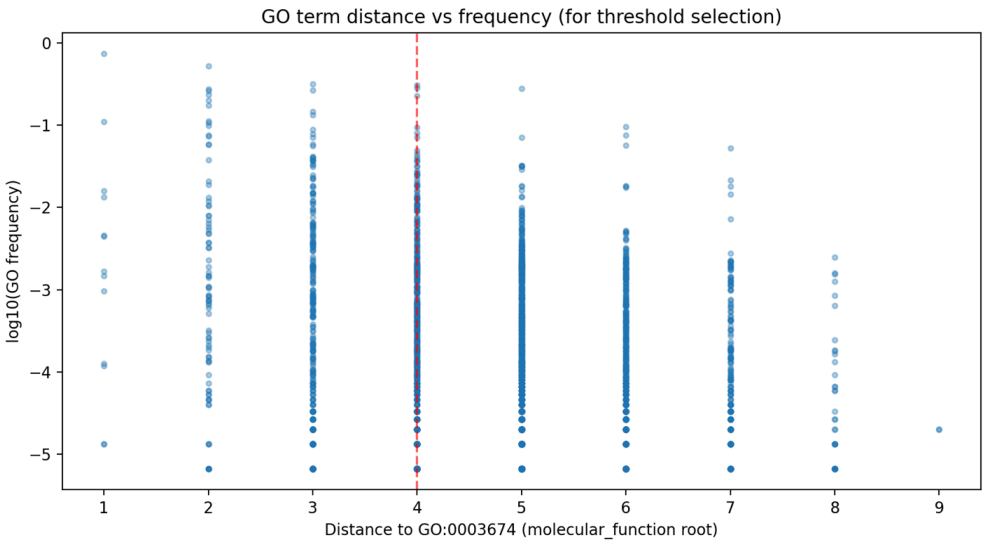
En comparant les prédictions pour tous les termes GO et toutes les protéines entre les deux jeux de donnés, les prédictions pour les séquences de cyanobactéries contiennent plus de prédictions à haute confiance. Les prédictions à 100 % de confiance dans les séquences shuffled pourraient correspondre aux termes GO présent dans presque tout les données d'entraînement, comme le terme GO pour ‘molecular function’ (GO:0003674), que notre modèle a pu apprendre par coeur. Cependant, nous avons vu que le modèle produit des prédictions de tous les niveaux de confiance pour les séquences shuffled. Cela veut dire qu'un simple cut-off de confiance ne permettra pas de distinguer les prédictions fiables de celles non-fiables.



Une simple heatmap de la clusterisation des prédictions permet de voir que les prédictions de termes GO dans les séquences shuffled ne sont pas toutes similaires. Même si les embeddings de TM-Vec occupent un espace similaire sur le UMAP, les prédictions ne sont pas les mêmes, et elles peuvent être similaires.

Nous avons continué en nous concentrant sur l’étude des termes GO prédits par le modèle. Dans cette partie, nous utilisons de nouveaux critères pour déterminer si une prédiction

compte comme un hit intéressant. D'abord, le critère de GO depth est appliqué, et le graphique ci-dessous est tracé afin de visualiser la distribution des termes GO :

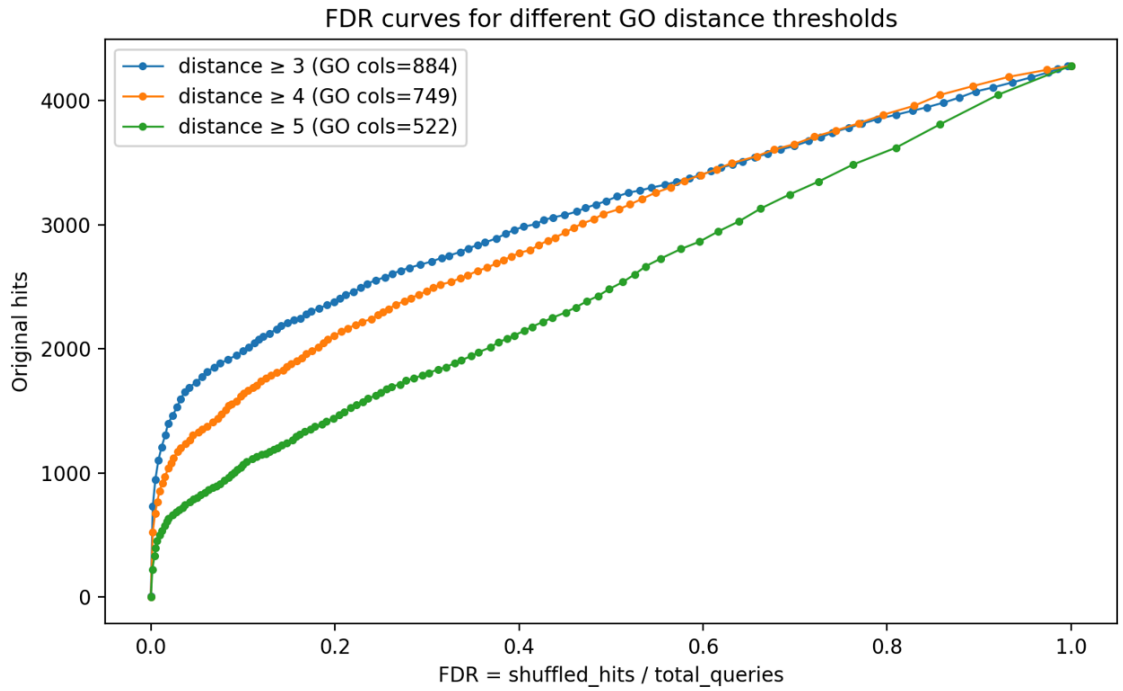


Ce graphique représente la relation entre la profondeur des termes GO (GO depth) et leur fréquence d'apparition dans la base de données. Pour chaque terme GO associé à la fonction moléculaire (MF), nous avons calculé sa distance minimale à la racine MF (GO:0003674), ce qui est définie comme la longueur du plus court chemin dans la hiérarchie GO. Cette distance correspond à la profondeur du terme GO. Ensuite, la fréquence de chaque terme GO est estimée à partir de la base de données d'annotations, comme le nombre d'occurrences du terme rapporté au nombre total de protéines annotées. Pour faciliter la visualisation, la fréquence est représentée sur une échelle logarithmique, sinon les points seraient confondus autour de 0.

Dans le graphique, l'abscisse correspond à la profondeur des termes GO, tandis que l'ordonnée indique la fréquence logarithmique de chaque terme dans la base de données. Les termes GO de faible profondeur (depth = 1 ou 2), qui sont proches de la racine, sont peu nombreux et présentent des fréquences élevées, parce que leur caractère est très général. La majorité des termes GO se situe aux profondeurs intermédiaires (depth = 3 et 4) qui constituent une région dense en annotations. Quant aux profondeurs plus élevées (depth ≥ 6), le nombre de termes diminue progressivement, ce qui reflète le caractère plus spécifique et plus rare de ces annotations fonctionnelles.

Enfin, nous pouvons déterminer le seuil de distance à partir de ce graphique. Les termes GO proches de la racine sont très fréquents mais peu informatifs sur le plan fonctionnel, parce qu'ils décrivent des catégories générales partagées par un grand nombre de protéines. En revanche, les termes GO très profonds sont plus spécifiques, mais leur faible représentation limite leur utilisation pour une analyse statistique robuste. Nous avons donc retenu un seuil de distance égal à 4, qui correspond à un compromis entre la spécificité fonctionnelle et la couverture des annotations. À partir de cette profondeur, les termes GO deviennent suffisamment spécifiques, en restant largement représentés dans la base de données.

Quand nous traçons les courbes, nous faisons également varier le seuil autour de 4, afin de voir l'effet de ce seuil sur le résultat de prédiction. Nous obtenons le graphique ci-dessous :



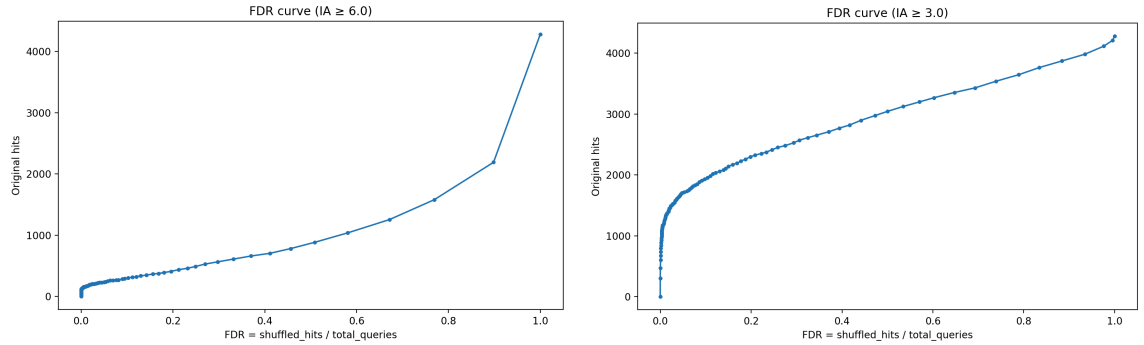
En comparant les trois conditions correspondant à des profondeurs GO ≥ 3, ≥ 4 et ≥ 5, on peut observer que la courbe bleue associée à la profondeur GO ≥ 3 présente une augmentation plus rapide du nombre de hits originaux aux faibles valeurs de FDR, ce qui traduit une meilleure couverture lorsque les critères sont plus permissifs. Ce comportement est attendu, car un seuil de profondeur plus faible conserve davantage de termes GO qui sont plus généraux et fréquents.

Cependant, les termes GO proches de la racine molecular function sont moins spécifiques et offrent une capacité de distinction fonctionnelle limitée. À l'inverse, l'augmentation du seuil de profondeur (GO ≥ 5) réduit le nombre de termes retenus mais favorise des annotations fonctionnelles plus spécifiques. Le seuil de GO ≥ 4 constitue un compromis équilibré entre performance statistique et spécificité fonctionnelle. Bien que moins de hits soient obtenus par rapport au seuil de GO ≥ 3 à un même FDR, les annotations retenues sont plus informatives.

Le choix du seuil de GO ≥ 4 est donc validé par l'analyse des courbes de FDR. À faible FDR dans le graphique, un nombre déjà important de séquences originales est annoté, tandis que les séquences shuffled sont majoritairement exclues. Cela indique que les prédictions du modèle sont statistiquement robustes après le filtrage des termes GO qui sont trop généraux. Le graphique n'est pas tracé pour fixer un seuil unique de FDR. Le but est d'utiliser cette courbe comme un outil de validation du modèle. Par exemple, pour un FDR d'environ 0,2, on obtient environ 2000 annotations fiables, ce qui constitue un compromis raisonnable entre la couverture des séquences et le niveau de confiance des prédictions.

En complément des seuils fondés sur la similarité structurelle (TM-score) et sur le GO depth, une approche alternative a été explorée en utilisant l'Information Accretion (IA) associée aux termes GO. L'IA mesure la spécificité informationnelle d'un terme GO dans la hiérarchie ontologique : plus la valeur d'IA est élevée, plus le terme est rare et informatif.

Dans ce cadre, un seuil minimal d'IA est fixé (par exemple IA ≥ 3 ou IA ≥ 6), et un hit est comptabilisé pour une séquence query si au moins un terme GO associé à ses voisins dépasse ce seuil. Cette stratégie permet de filtrer les annotations trop générales et de se concentrer sur des prédictions fonctionnelles plus spécifiques. Comme précédemment, les graphiques suivants sont tracés :



La courbe obtenue avec un seuil strict (IA ≥ 6.0) montre que, pour des valeurs faibles de FDR, le nombre de hits reste très limité. Une augmentation significative du nombre de prédictions n'apparaît que lorsque le FDR devient élevé, indiquant que la majorité des hits supplémentaires correspondent alors à des annotations susceptibles d'être dues au hasard. Ce résultat suggère que, bien que les termes GO très informatifs soient particulièrement intéressants en biologie, le signal exploitable reste encore faible, en raison d'un compromis défavorable entre spécificité des annotations et contrôle du taux de faux positifs.

En abaissant le seuil à 3.0, la courbe FDR présente une augmentation beaucoup plus rapide du nombre de hits à partir des faibles valeurs de FDR. Cela indique qu'un plus grand nombre de séquences peuvent être associées à des termes GO tout en maintenant un niveau de faux positifs relativement contrôlé. Toutefois, cette amélioration quantitative s'accompagne d'une perte de spécificité fonctionnelle, les termes GO inclus à ce seuil étant plus généraux et moins discriminants. Ainsi, le gain en nombre de prédictions ne se traduit pas nécessairement par une meilleure précision fonctionnelle.

Perspectives

Améliorer la base d'entrainement

Comme vu avec le UMAP, certains des embeddings des séquences des données d'entraînement, de test et de validation (d'Uniprot) sont mappées sur les séquences shuffled. Supprimer les séquences concernées de la base de données initiale pourraient permettre d'avoir des résultats plus riches et précis. Cela ne permettrait pas de trouver les termes GO des séquences réelles semblables au shuffled, mais cela permettrait d'assurer des résultats

sûrement plus fiables. De plus, les résultats de termes GO ont été obtenus que pour les séquences de cyanobactéries d'une taille inférieure ou égale à 1200 (d'après les séquences récupérées sur la base de données Uniprot). C'est pourquoi, pour moins de 100 séquences de Cyanorak (qui ont une longueur supérieure à 1200) les termes GO n'ont pas été cherché.

Continuation de PlasmoFP

L'article et le code de PlasmoFP contiennent plusieurs étapes après le premier entraînement du modèle qui pourraient permettre d' améliorer nos résultats. Une prochaine étape serait d'entraîner un modèle de deep-ensemble, c'est-a-dire entraîner plusieurs modèles sur les mêmes données et combiner leurs prédictions. Cela pourrait améliorer les résultats sur les séquences shuffled, qui pourraient varier plus au niveau des confiances accordées aux termes GO. Le code initial de PlasmoFP contient aussi une analyse d'incertitudes de termes GO, qui permet d'ajuster les scores pour choisir des seuils individuels à chaque terme.

Autres modèles et architectures

Les résultats présentés dans ce projet montrent l'intérêt d'une approche basée sur des représentations vectorielles de protéines, en soulignant également la nécessité de replacer ces résultats dans un cadre comparatif plus large. Une perspective nécessaire serait donc de confronter les performances obtenues avec PlasmoFP à celles d'autres méthodes existantes de prédiction des termes de Gene Ontology. En particulier, il serait pertinent de comparer nos résultats à d'autres approches telles que DeepGO, qui utilisent des modèles supervisés entraînés spécifiquement pour l'annotation GO. Ce type de comparaison permettrait d'évaluer de manière concrète si l'approche proposée apporte un avantage en termes de précision des prédictions et de contrôle du FDR, ou si elle est plutôt une solution complémentaire aux méthodes déjà existantes.

Enfin, pour garantir une comparaison équitable entre les différentes approches, il serait utile d'adopter des critères d'évaluation communs, tels que l'analyse des courbes FDR ou l'utilisation de seuils basés sur l'information accretion des termes GO. Ce cadre d'évaluation permettrait de mieux apprécier les avantages et les limites de chaque méthode et d'orienter le choix du modèle en fonction des objectifs du projet.