Université Paris Cité Master 2 Biologie-Informatique 2021-2022

**Rapport de Stage M2**

**Histoire évolutive des protéines candidates à la régulation post-transcriptionelle des génomes chloroplastiques et mitochondriaux des diatomées**

*Goulancourt Rebecca*

*Github : https://github.com/Rebbekkah/Stage\_M2.git*

*Mots clés : Chloroplaste, protéine, adressage, régulation post-transcriptionelle, alpha-solénoïde, prédiction, modèle de prédiction « Random Forest ».*

*Tutrice : Ingrid Lafontaine. Site : IBPC - CNRS, 13 rue Pierre et Marie Curie,*

*75005 Paris*

*SOMMAIRE*

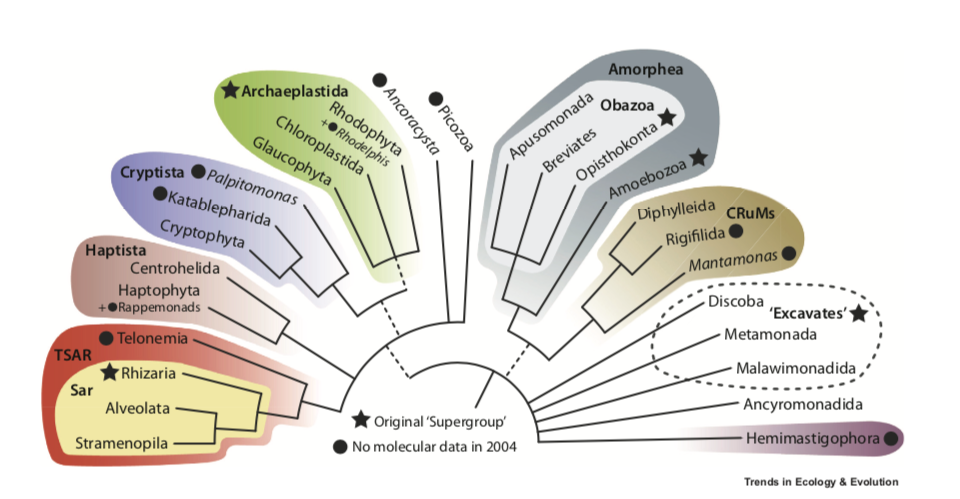
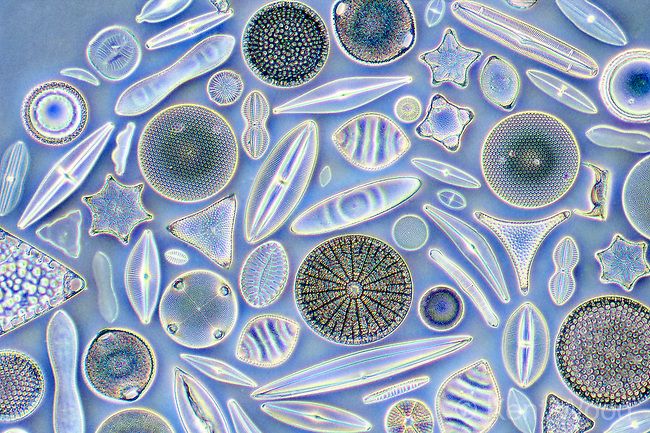
1. Introduction & état de l’art

Les diatomées sont des organismes marins, microalgues eucaryotes photosynthétiques de diverses formes. Leur taille peut varier de 2 µm à 1mm et en tant qu’espèce photosynthétique elles sont dotées de mitochondries et surtout de chloroplastes. Ce chloroplaste leur vient d’une endosymbiose secondaire. L’endosymbiose secondaire étant l’internalisation d’une micro algue elle même issue d’une endosymbiose primaire. Les cellules issues de l’endosymbiose primaire, comme les Archaeplastida, proviennent de l’internalisation (probablement par phagytose) d’une cyanobactérie par une cellule ancestrale primaire hétérotrophe. La cyanobactérie se différencie ensuite en chloroplaste et la cellule acquiert une capacité photosynthétique, devenant ainsi autotrophe. Les diatomées faisant parti de la famille des *Stramenopila* (SAR) découlent de l’internalisation d’une micro algue rouge (Rhodophyte) autotrophe possédant un chloroplaste par une autre cellule hétérotrophe, obtenant ainsi des caractéristiques photosynthétiques. Le chloroplaste, au même titre que la mitochondrie, est un organite. Ils proviennent donc de l’endosymbiose de bactéries ancestrales et sont conservées au cours de l’évolution. Ainsi elles possèdent donc du matériel génétique qui s’exprime et leur sont propre.

Dont algues vertes

A

B



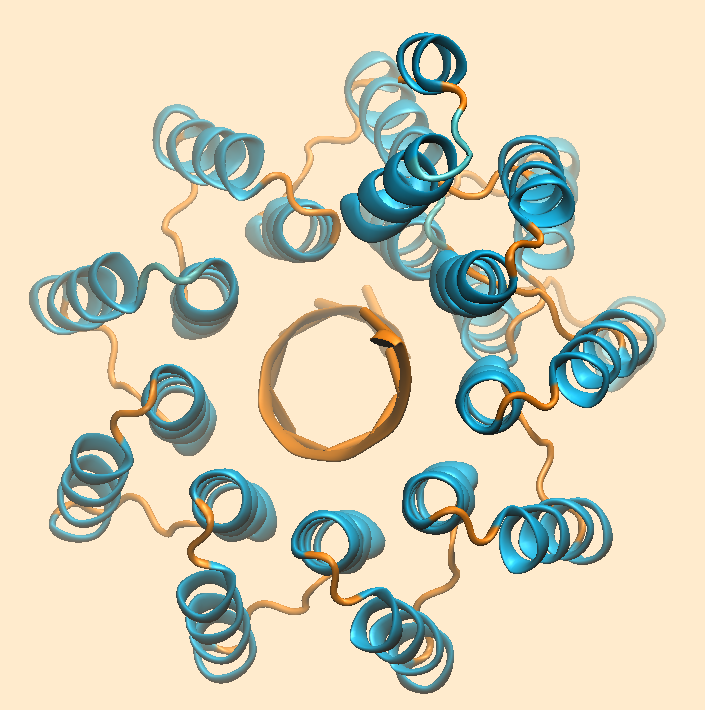
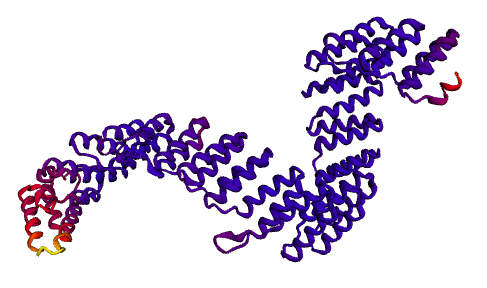
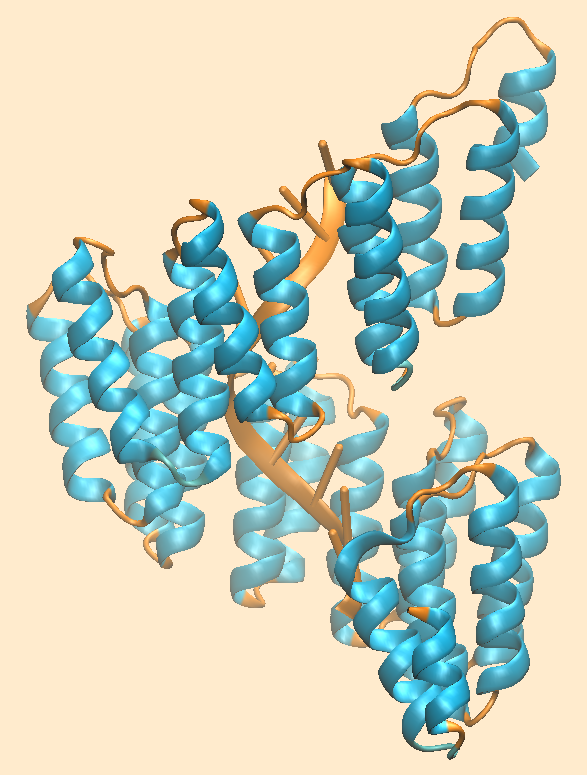
Endosymbiose 1aire

Dont diatomées

Endosymbiose 2aire

**Figure 1 :** (A) Diatomées observées en microscopie à contraste de phase. (B) Arbre phylogénétique des eukaryotes [Burki & al.]. Événement d’endosymbiose primaire avec une cyanobactérie.

La régulation post-transcriptionelle du chloroplaste est connue chez les eucaryotes photosynthétiques issus d’une endosymbiose primaire, mais beaucoup moins chez les SAR dont font partis les diatomées. Toutefois ces organismes étant proches évolutivement parlant, nous utiliseront nos connaissances des Archaeplastida dans notre étude. Ainsi chez les Archaeplastida ce type de mécanisme s’effectue principalement par des protéines possédant une structure particulière appelée alpha-solénoïde. Elles sont aussi connues pour interagir particulièrement avec l’ARN messager chloroplastique dans des processus de maturation, stabilisation, épissage, translation et dégradation génétique. Ces protéines en alpha solénoïdes sont adressées aux organites : mitochondrie et chloroplaste. Dans la suite de ce rapport du désignerons par ROGEs [Boulouis & al.] les protéines en alpha-solénoïdes adressées aux organites. En effet, les protéines en alpha solénoïdes de par leur nature régulatrice font donc parties des protéines ROGEs, pour Regulator Organelle Gene Expression. Les motifs en alpha solénoïdes sont constitués d’une répétition d’hélices alpha en antiparallèle et conférant leur structure caractéristique aux protéines. Un motif de répétition constitue alors deux hélices antiparallèles. De part leur forme, les protéines en alpha solénoïdes se fixent sur l’ARNm chloroplastique pour le réguler. En effet, l’ARNm se positionne du côté 5’ au sein de la cavité concave formée par la structure en hélice alpha pour se lier à la protéine. De plus, il semblerait que la méthode de reconnaissance de l’ARNm soit due à une spécificité des bases azotées et d’une reconnaissance dite « one repeat :one-nucleotide ». Cela signifie que la reconnaissance nucléotidique permet à un nucléotide de se fixer entre deux motifs d’hélices.

C

A

B

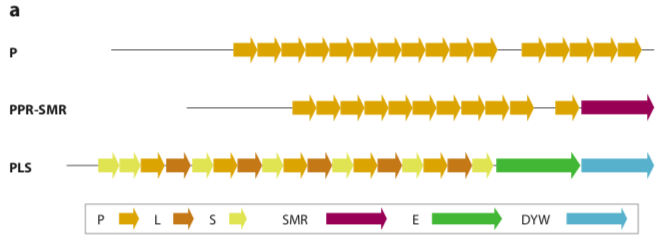
**Figure 2 :** (A) Protéine en alpha solénoïde. (B) Protéine en alpha solénoïde liée à un ARNm du côté 5’, le carré noir montrant 1 motif de répétition d’hélice alpha en antiparallèle. (C) De même que la figure 2B, la flèche représentant le positionnement des nucléotides entre deux motifs : c’est ce que l’on appelle la reconnaissance par « one repeat : one nucleotide ». © Céline Cattelin, IBPC.

Parmi les protéines en alpha solénoïdes on distingue trois types de protéines classifiées selon les répétitions au sein de leur séquence :

* les TPR, répétitions de 34 acides aminés (tetracopeptide repeat), spécialisées dans les interactions protéine-protéine.
* les PPR, répétitions de 35 acides aminés (pentatricopeptide repeat), ayant un rôle dans l’interaction protéine-ARN.
* les OPR, répétitions de 38 acides aminés (octatricopeptide repeat) dont la fonction consiste aux interactions protéine-ADN.

Même si les OPR et TPR restent peu étudiées et de ce fait peu connues, les PPR sont les plus caractérisées des trois. En effet, les PPR sont présentes près de 400 fois plus chez les plantes que chez les microalgues. Les PPR sont elles mêmes classifiées en trois sous-groupes : P-PPR, PPR-SMR et PLS-PPR.

*Editing, protection et repliement*



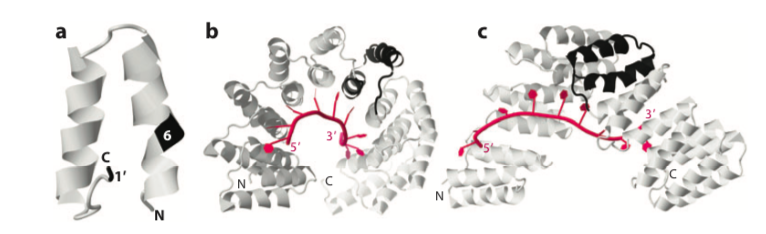
*Stabilisation, clivage, épissage*

*Endonucléase*

**Figure 3 :** Les différents motifs présents chez les PPR [Barkan & Small.].

Ainsi les P-PPR ne sont constituées que de motifs P leur conférant des activités d’editing, de protection face aux nucléases et de repliement (celui-ci jouant un rôle dans la traduction). Les PPR-SMR de motifs P et d’un motif SMR. Ce motif SMR (MutS related domain) se situant en fin de séquence (C-terminal) possède un rôle d’endonucléase. Pour ce qui est des PLS-PPR elles sont constituées de différents motifs : P, L, S ou encore E et DYW. Le motif DYW (Asp-Tyr-Trp) est aussi appelé le « cytidine deaminase-like domain » et permet ainsi la désamination des cytidines en uraciles. Toutefois la fonction exacte des domaines E et DYW ne sont pas véritablement connus chez les PPR mais nous savons que les PLS-PPR possèdent un rôle de stabilisation, clivage et d’épissage.

De plus il a été constaté chez les alpha-solénoïdes en PPR et en OPR que la reconnaissance nucléotidique se fait par l’intermédiaire du premier et du sixième acide aminé de la protéine régulatrice.



C

A

B

**Figure :** Structure d’une PPR (PRORP1 RNaseP chez A.Thaliana). (A) Un seul motif PPR de la protéine. Les positions 1’ et 6’ sont coloriées en noir et déterminent la spécificité de liaison des nucléotides aux motifs PPR. (B) & (C) L’ARNm se fixe au sein de la protéine et les nucléotides se lient entre chaque motif. Un motif est surligné en noir. [Barkan & Small]

Parmi les OPR, il semblerait qu’il existe des motifs appelés RAP impliqué dans la reconnaissance des ARNm ainsi que de l’ADN. Ainsi de par le manque de connaissances de ces protéines il est difficile de pouvoir reconnaître avec exactitude les protéines ROGEs chez les diatomées et les algues rouges en général. De ce fait nous voulons pouvoir les reconnaître au sein de protéomes de diatomées afin de pouvoir les annoter et les étudier plus précisément. Le but de ce projet est donc d’automatiser la reconnaissance de protéines candidates à la régulation post-transcriptionelle chez les diatomées par machine learning, sur la base de leurs propriétés structurelles et physico-chimiques.

Dans un premier temps nous utiliserons la méthode sur les organismes de *Chlamydomonas reinhardtii et d’Arabidopsis Thaliana* pour la tester.

1. Matériel et méthodes

Les codes permettant de répondre aux questions et problématiques ont été rédigés essentiellement en python (version 3.9), puis en R.

|  |  |
| --- | --- |
| **Python** | **R** |
| *Panda, Numpy, os, glob, seaborn, basename, matplotlib, statistics, glob, operator, itemgetter, sklearn, TSNE scipy, dist, shapiro, f, RandomForestClassifier, GridSearchCV, confusion\_matrix* | *Protr, Biostrings, optparse, stringr, M3C, Rtsne, RcolorBrewer, seqinr* |

**Table 1 :** Modules utilisés pour les analyses bio-informatiques.

Nous cherchons à pouvoir identifier des protéines en alpha solénoïdes impliquées dans la régulation post-transcriptionelle sur la base de leurs propriétés structurales et physico-chimiques. Nous savons que ces protéines ont des caractéristiques communes :

* Elles sont adressées aux organites : chloroplaste et/ou mitochondrie.
* Elles ne possèdent pas de domaines transmembranaires. Si un tel domaine est détecté avant le 68e acide aminé il pourrait en fait correspondre au peptide d’adressage qui permet aux protéines codées dans le noyau et traduites dans le cytoplasme d’être importées aux organites.
* De par leur structure en hélice α elles possèdent des paires d’hélices α. Ainsi elles ont dans le code génétique des répétitions de séquence caractéristiques de ces hélices.

Pour déterminer les possibles candidats à la régulation post-transcriptionelle nous disposons d’un protocole au préalablement conçu au laboratoire par la doctorante Céline Cattelin (🡪 peut être mettre en annexe). Il permet d’identifier la plupart de ces protéines à l’aide de différentes étapes et de logiciels de prédictions utilisés en stand alone.

Peptide d’adressage

3’

*Absence de domaine TM*

5’

1. Détection domaines TM

Protocole

*Présence de motifs répétés*

3’

68

5’

2. Détection motifs répétés

*Présence de paires d’hélice*

L

L

α

α

α

α

3’

68

5’

4. Prédiction localisation intracellulaire

3. Détection paires d’hélices

**Organites** (mitochondrie, chloroplaste)

**Figure 5 :** Protocole mis en place pour identifier les protéines en alpha solénoïdes. Si une protéine passe tout ces filtre alors elle est candidate pour faire partie des protéines ROGEs. La présence d’hélice α est détectée par l’intermédiaire d’acides aminés « linker » (L) qui relient deux hélices entre elles.

Ces outils, leurs prédictions et les informations que l’on souhaite récupérer sont visibles dans la partie résultats (table 2). Les protéines passant les différents filtres seront considérés comme des protéines en alpha solénoïde et candidates à la régulation post-transcriptionelle.

Le but final de notre étude sera de pouvoir distinguer les protéines ROGEs en utilisant les différents filtres établis sur la base de leur capacité à distinguer les alpha solénoïdes connues. Pour cela nous construirons un modèle prédictif qui pourra déterminer si une protéine est candidate ou non à la régulation post-transcriptionelle chloroplastique. Chaque logiciel prends en entrée un protéome, et a un output (ou sortie) formaté qu’il sera ensuite possible de parser (parcourir à l’aide d’un script) pour récupérer les informations qui nous intéressent.

Nous utilisons donc 7 logiciels :

Le logiciel TMHMM permet de prédire la présence et la position de segments transmembranaires. Il fonctionne sur le modèle de chaînes cachées de Markov-Nikov[[1]](#footnote-1). Il permet ainsi de déterminer si une protéine est transmembranaire ou non, nos protéines d’intérêt ne l’étant pas. L’output de TMHMM est un fichier qui nous donne pour chaque protéine du protéome le nombre de segments transmembranaires prédits ainsi que leur positions.

Targetp2 quant à lui est un modèle prédictif de deep learning, entraîné et testé sur des modèles de séquences construits à partir de la matrice de substitution BLOSUM62. Il sert à détecter des séquences dites « signales » dans l’adressage protéique. Ces séquences sont propres à un adressage qui peut être dans le cas qui nous intéresse mitochondrial ou chloroplastique. En sortie, Targetp2 écrits un fichier pour le protéome d’entrée qui contient pour chaque protéine l’adressage qui a été déterminé.

Le logiciel Ard2 (Alpha-rod Reapeat Detector) rend possible l’identification de sites de linker entre hélices alpha, permettant ainsi la détection de protéines possédant des structures en alpha solénoïdes en utilisant un réseau neuronal. Ce logiciel nous donne pour chaque acide aminé de chacune des séquences la probabilité associée d’être un linker.

RADAR (Rapid Automatic Detection adn Alignment of Repeats) permet la détection de motifs répétés au sein de séquences protéiques. Ici, cela nous permettra de potentiellement repérer la présence d’hélices α qui sont des structures aux motifs répétés. Ce logiciel fonctionne sur un algorithme capable d’identifier des répétitions de motifs de différents types. En sortie radar écrit un fichier contenant principalement pour chaque séquences du protéome : le nombre de répétitions trouvées, leurs positions et leur séquence.

Quant à Deeploc, Wolfpsort et Localizer ils permettent la prédiction d’adressage d’une protéine. En effet Deeploc distingue 10 localisations cellulaires différentes[[2]](#footnote-2) et sur la base de la séquence permet d’associer une probabilité d’adressage pour chaque localisation à une protéine. Il fonctionne grâce à des réseaux neuronaux récurrents.

Wolpsort utilise la composition en acide aminé qu’il convertit en vecteurs pour être ensuite classifiés par méthode k-nearest neighbor. C’est un algorithme supervisé de machine learning qui peut être utilisé à des fins de classification ou de régression. L’algorithme permet donc de déterminer la localisation cellulaire d’une protéine par l’intermédiaire de sa proportion en acide aminé. En sortie wolfpsort nous rends un fichier contenant un identifiant protéique par ligne dont l’adressage est noté par un score puis ordonné de manière décroissante, le total des scores n’étant pas fixe.

Enfin, l’algorithme de Localizer est une méthode de machine learning qui a été entraîné pour prédire la localisation des protéines chez les plantes. L’adressage vers le chloroplaste ou la mitochondrie est prédite en déterminant la présence de peptide signal (ou peptide de transit) tandis que la localisation vers le noyau est prédite par un ensemble de signaux peptidiques nucléaires (NLSs). Pour ce qui est du fichier de sortie, on obtient essentiellement par ligne l’identifiant protéique, et la première colonne correspondant à l’adressage au chloroplaste. Si la protéine est identifiée comme étant ciblée pour cet organite alors la première colonne sera marquée d’un ‘Y’.

Tous ces logiciels ont été utilisés en stand alone et non en libre service sur internet. C’est-à-dire que nous avons utilisé directement les codes et script sources que nous avons lancé dans le terminal à l’aide des indications des README pour chaque logiciels.

🡪 est ce que je dois vraiment parler des outputs des logiciels ??

Tous ces logiciels ont été utilisés en stand alone et non en libre service sur internet. C’est-à-dire que nous avons utilisé directement les codes et script sources que nous avons lancé dans le terminal à l’aide des indications des README pour chaque logiciels.

Nous avons donc fait tourner d’abord TMHMM sur ces protéomes et récupéré en effectuant un parsing des output les identifiants des protéines ne possédant pas de domaines transmembranaires après le 68e acide aminé (si il y en a avant cela correspond à un peptide d’adressage qui sera clivé et perdu à l’entrée de l’organite). Ensuite nous avons reconstitué de nouveaux protéomes avec les séquences des protéines sélectionnées à l’issu du filtrage. Ce sont ces nouveaux protéomes qui ont ensuite été donnés comme input aux autres logiciels. Il est donc important de commencer par utiliser TMHMM, même si l’ordre d’utilisation des autres logiciels reste négligeable. Chacun des autres logiciels a donné des outputs que l’on a filtré pour ne garder que l’essentiel, information visible ci-dessus dans le tableau récapitulatif (annexe 2).

Pour ce qui est du parsing de targetp2, nous avons récupéré les résultats d’adressage prédit pour chaque protéine, que nous avons stocké dans un dictionnaire. Nous avons fait de même avec Deeploc et Localizer, et récupéré des valeurs de probabilité d’adressage lorsque cela était nécessaire.

En première sortie ard2 réécrit le protéome pour le formater. Ensuite il faut redonner ce premier output en entrée à ard2 pour que l’output final soit accessible. Celui-ci donne pour chaque séquence la probabilité de chaque acide aminé d’être un linker. Ainsi nous sélectionnons les acides aminés dont la probabilité est supérieure à 0.10. De plus, si l’on dispose d’une « région de linker » nous considérons qu’il faut sélectionner l’acide aminé avec la plus grande probabilité de linker. Pour cela nous avons utilisé le principe de fenêtre glissante. Cette fenêtre parcourt chaque acide aminé et son résultat par groupe de 6 acides aminés. Au sein de cette fenêtre nous prenons l’acide aminé qui possède la plus forte probabilité lorsqu’elle est supérieure à 0.10, ainsi que sa position au sein de la séquence.

Pour le logiciel radar il a fallu se poser la question du chevauchement des répétitions, auxquels cas il faut prendre la plus grande longueur de répétition possible. En effet radar peut tout à fait détecter des répétitions dont les positions au sein de la séquence se chevauchent. Toutefois dans les cas où des répétitions ont été détectées au sein même d’autres répétitions il a fallu les supprimer du jeu de données.

Répétition 1

C-terminal

N-terminal

***Cas 1***

Répétition 2

N-terminal

C-terminal

Répétitions prise en compte

Répétition 1

C-terminal

N-terminal

Répétition 2

***Cas 2***

C-terminal

N-terminal

Répétitions prise en compte

(la plus grande des deux)

**Figure 6 :** Exemple de répétitions chevauchantes trouvées par Radar. Les répétitions 1 et 2 sont deux types de répétitions différentes détectées. 🡪 si rentre pas dans résultats alors annexes !!

Ainsi il existe deux cas de chevauchements possibles. Dans le premier il y a un chevauchement entre le début et la fin des répétitions. Dans ce cas la répétition prise en compte correspond au début de la répétition 1 et la fin de la répétition 2. Dans le second cas on en prend en compte que la répétition qui englobe la seconde.

Pour Wolfpsort nous avons choisi de récupérer les scores associés aux adressages aux organites lorsqu’il y avait une prédiction de ce type. Ces scores ont ensuite été additionnés puis normalisé sur chaque séquence en le divisant par le score total, différent pour chaque protéine.

Finalement, pour regrouper tous nos résultats nous avons fabriqué une dataframe contenant par lignes les identifiants protéiques et pour chaque colonne les résultats des parsing des différents logiciels. Enfin en dernier lieu nous avons décidé de modifier les types de données de la dataframe finale, dont un exemple est visible ci-dessous et en annexe :

Ce sont ces résultats que nous allons utiliser pour entraîner notre modèle.

Pour notre modèle de prédiction nous utiliserons aussi des caractéristiques comme la fréquence en acide aminé, mais aussi les valeurs d’ACC sur Z-scales de la protéine. L’ACC (Auto Cross Correlation) est une mesure entre dans notre cas ce que l’on appelle des Z-scales ou Z-scores. Ces Z-scales sommairisent les caractéristiques physico-chimiques d’une protéine. Pour chacun des 20 acides aminés on distingue trois valeurs de Z-scales :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Z1** | **Z2** | **Z3** |
| Caractéristique | Hydrophobicité | Encombrement stérique | Propriétés électroniques |

**Table 4 :** Z-scores et les propriétés qu’elles caractérisent.

Ces Z-scales permettent d’expliquer la variabilité de structure et d’activité des peptides. L’ACC correspond donc à la covariance entre Z-scales d’un ou plusieurs acides aminés pour un lag donné. Nous avons choisit de prendre un lag (ou fenêtre) de 4 car cela est le plus conforme à un tour d’hélice de 3.6 acides aminés. Comme un acide aminé se caractérise par 9 valeurs d’ACC, pour un lag de 4 un peptide sera caractérisé par 36 valeurs. Un graphe explicatif et les formules permettant de calculer les ACC seront donnés en annexe.

Ainsi pour un même facteur (z1 – z1, z2 – z2 et z3 – z3) on obtient 3\*r = 3\*4 = 12 valeurs d’ACC. Tandis que pour des facteurs différents (z1 – z2, z1 – z3, z2 – z3, z3 – z1, …) on a 6\*r = 6\*4 = 24 ACC soit 12+24 = 36 termes d’ACC. Ainsi les ACC combinent les informations de l’auto-cross variance entre mêmes facteurs à chaque position et la cross variance entre deux différents Z-scales à chaque positions. Les ACC constituent donc une approche physico-mathématique quant à la description d’un peptide.

*Mise en œuvre du modèle :*

Pour ce qui est du modèle prédictif, nous avons fait le choix d’un modèle de type « Random Forest ». Le Random Forest est un algorithme utilisé en machine learning se basant sur le principe d’arbres décisionnels. Ici, les nœuds correspondraient aux différentes étapes du protocole de la figure 3 qui sont nos critères de décision (ou features). La suite de choix du modèle nous amènerait à deux décisions possibles : protéine ROGEs ou non. La Random Forest est en réalité un ensemble de plusieurs arbres décisionnels pour combiner les résultats afin qu’ils soient plus précis. La RF peut être utilisée à des fins de classification ou de régression. Dans notre cas nous allons donc l’utiliser pour classifier nos protéines. Chaque arbre décisionnel est entraîné sur un sous-ensemble de la dataset (échantillon) et donnera son propre résultat. Ensuite dans le cas d’une classification toutes les décisions sont combinées et le résultat final sera le celui qui aura eu la plus grande chance d’être piochée aléatoirement parmi toutes les décisions de chacun des arbres. Tandis que pour une approche de régression le résultat constituera la moyenne des prédictions à travers tous les arbres de l’ensemble de la RF. Le RF possède divers avantages comme une robustesse au sur-apprentissage grâce à sa « forêt » d’arbres décisionnels, mais surtout à l’issue de l’apprentissage il nous sera possible de déterminer le poids de chacun des descripteurs peptidiques. Ainsi nous pourrons connaître quels sont les caractéristiques permettant de distinguer une protéine ROGE d’une autre selon le modèle.

En premier lieu deux protéomes ont été fournis. Ces protéomes contiennent les séquences au format .fasta de protéines que l’on sait être en alpha solénoïde (pour le protéome de témoin positif) et de protéines qui ne le sont pas (pour le protéome de témoin négatif). Ces deux protéomes ont été remplis avec des protéines des organismes de Chlamydomonas reinhadtii, Arabidopsis Thaliana et un protéome de diatomés. Nous ferons tourner les logiciels cités plus haut sur ces deux jeux de données protéomiques qui serviront à faire apprendre notre modèle. Ainsi nous récupérerons une dataframe contenant en index les identifiants pour chaque protéine et en colonne le résultat du parsing des outputs pour chacun des logiciels. Nous calculerons aussi sur ces séquences les valeurs d’ACC et de fréquences d’acides aminés. Ensuite nous disposerons de 6 protéomes de différentes diatomées sur lesquels nous essaierons de déterminer quelles sont les protéines ROGEs afin de tester notre modèle. Ces protéomes sont constitués de *Cyclotella Cryptica, Fistulifera solaris, Phaedodactylum Tricornutum, Pseudo – nitzchia Multistriata, Thalassossira Oceanica et Tahlassossira Pseudonana*.

Pour ce qui est de l’apprentissage, test et validation du modèle nous utiliserons nos données comme suivant :

10%

90%

Dataset

***+ & -***

Apprentissage

Validation

20%

70%

Test

**Figure 7 :** Méthode d’échantillonnage des données positives et négatives. Les 10% de l’échantillon de validation permettront le calcul des performances du modèle.

Les étapes d’apprentissage et de test du modèle nous permettrons de réguler au mieux les paramètres du modèle (ou hyperparamètres) tandis que l’échantillon de validation nous permettra de vérifier le pourcentage d’erreur que pourra faire notre modèle.

Enfin nous pourrons déterminer l’efficacité du modèle par sa précision (taux de bonnes prédictions), sensibilité (probabilité d’avoir un vrai positif) et spécificité (probabilité d’avoir un vrai négatif). Ces informations seront résumées dans ce que l’on appelle une matrice de confusion. Le but est donc de maximiser ces performances afin d’avoir le meilleur modèle prédictif possible.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Précision** | **Sensibilité** | **Spécificité** |
| Définition | Taux de bonnes prédictions | Probabilité d’avoir un vrai positif | Probabilité d’avoir un vrai négatif |
| Formule |  |  |  |

**Table :** Formules du calcul des performances du modèle. Plus la précision est élevée plus le modèle est capable de prédire avec exactitude la nature de la protéine (ROGE ou non).

À la suite de cela si nous obtenons des performances satisfaisantes nous ferons tourner le modèle sur les protéomes cibles : 1. *Chlamydomonas reinhardtii & Arabidopsis Thaliana*, 2. *Les protéomes de diatomées.* Nous pourrons ainsi déterminer quelles protéines appartiennent à la catégorie des ROGEs.

1. Résultats
2. Chlamydomonas reinhardtii & Arabidopsis Thaliana

Mettre ici le traitement des outputs des logiciels

Le traitement des sorties des logiciels a été comme suivant :

|  |  |  |
| --- | --- | --- |
| **Outil** | **Prédiction** | **Information** |
| TMHMM | Structure transmembranaire | Identifiants des protéines sans domaine transmembranaire (après le 68e acide aminé) |
| Radar | Répétitions de séquences | Nombre de répétitions et proportion de la séquence qui possède des répétitions |
| Ard2 | Acide aminés linker d’hélice α | Nombre de linker, probabilités d’être linker |
| Targetp2 | Présence de pré-séquence en Nter (peptide d’adressage) | Adressage[[3]](#footnote-3) |
| Deeploc | Adressage | Probabilité d’adressage pour la mitochondrie et le chloroplaste |
| Wolfpsort | Adressage | Identifiants des protéines adressées à la mitochondrie ou au chloroplaste |
| Localizer | Adressage | Identifiants des protéines adressées au chloroplaste |

**Table 2 :** Tableau récapitulatif des outils utilisés et des informations recueillies après parsing des outputs.

Ainsi pour le logiciel TMHMM nous avons donc fait tourner d’abord ces protéomes et récupéré en effectuant un parsing des output les identifiants des protéines ne possédant pas de domaines transmembranaires après le 68e acide aminé (si il y en a avant cela correspond à un peptide d’adressage qui sera clivé et perdu à l’entrée de l’organite). Ensuite nous avons reconstitué de nouveaux protéomes avec les séquences des protéines sélectionnées à l’issu du filtrage. Ce sont ces nouveaux protéomes qui ont ensuite été donnés comme input aux autres logiciels. Il est donc important de commencer par utiliser TMHMM, même si l’ordre d’utilisation des autres logiciels reste négligeable.

Pour ce qui est du parsing de targetp2, nous avons récupéré les résultats d’adressage prédit pour chaque protéine, que nous avons stocké dans un dictionnaire. Nous avons fait de même avec Deeploc et Localizer, et récupéré des valeurs de probabilité d’adressage lorsque cela était nécessaire.

En première sortie ard2 réécrit le protéome pour le formater. Ensuite il faut redonner ce premier output en entrée à ard2 pour que l’output final soit accessible. Celui-ci donne pour chaque séquence la probabilité de chaque acide aminé d’être un linker. Ainsi nous sélectionnons les acides aminés dont la probabilité est supérieure à 0.10. De plus, si l’on dispose d’une « région de linker » nous considérons qu’il faut sélectionner l’acide aminé avec la plus grande probabilité de linker. Pour cela nous avons utilisé le principe de fenêtre glissante. Cette fenêtre parcourt chaque acide aminé et son résultat par groupe de 6 acides aminés. Au sein de cette fenêtre nous prenons l’acide aminé qui possède la plus forte probabilité lorsqu’elle est supérieure à 0.10, ainsi que sa position au sein de la séquence.

Pour le logiciel radar il a fallu se poser la question du chevauchement des répétitions, auxquels cas il faut prendre la plus grande longueur de répétition possible. En effet radar peut tout à fait détecter des répétitions dont les positions au sein de la séquence se chevauchent. Toutefois dans les cas où des répétitions ont été détectées au sein même d’autres répétitions il a fallu les supprimer du jeu de données.

Répétition 1

C-terminal

N-terminal

***Cas 1***

Répétition 2

N-terminal

C-terminal

Répétitions prise en compte

Répétition 1

C-terminal

N-terminal

Répétition 2

***Cas 2***

C-terminal

N-terminal

Répétitions prise en compte

(la plus grande des deux)

**Figure 6 :** Exemple de répétitions chevauchantes trouvées par Radar. Les répétitions 1 et 2 sont deux types de répétitions différentes détectées. 🡪 si rentre pas dans résultats alors annexes !!

Ainsi il existe deux cas de chevauchements possibles. Dans le premier il y a un chevauchement entre le début et la fin des répétitions. Dans ce cas la répétition prise en compte correspond au début de la répétition 1 et la fin de la répétition 2. Dans le second cas on en prend en compte que la répétition qui englobe la seconde.

Pour Wolfpsort nous avons choisi de récupérer les scores associés aux adressages aux organites lorsqu’il y avait une prédiction de ce type. Ces scores ont ensuite été additionnés puis normalisé sur chaque séquence en le divisant par le score total, différent pour chaque protéine.

Finalement, pour regrouper tous nos résultats nous avons fabriqué une dataframe contenant par lignes les identifiants protéiques et pour chaque colonne les résultats des parsing des différents logiciels. Ainsi en combinant les descripteurs des logiciels de prédictions, ACC et fréquences d’acides aminés nous obtenons une dataframe de 51 755 lignes et 66 ?? colonnes.

Ce sont ces résultats que nous allons utiliser pour entraîner notre modèle et permettre sa prédiction.

Aussi, pour être sûr de la pertinence des logiciels choisis nous avons décider de d’étudier comment les résultats se comportaient selon les jeux positifs et négatifs. Nous avons donc d’abord regardé les boxplots de la répartition des résultats (visibles en annexes) et effectué des tests de comparaison de moyennes entre les deux dataset. Pour chacun des logiciels, les test ont été significatifs : il y a bien une différence de résultats et donc de structure/conformation au sein des protéines positives et négatives.

Pour ce qui est du modèle en soit nous avons décider d’utiliser les paramètres suivant qui permettaient d’avoir les meilleures performances possibles : 🡪 disponible en annexe ?

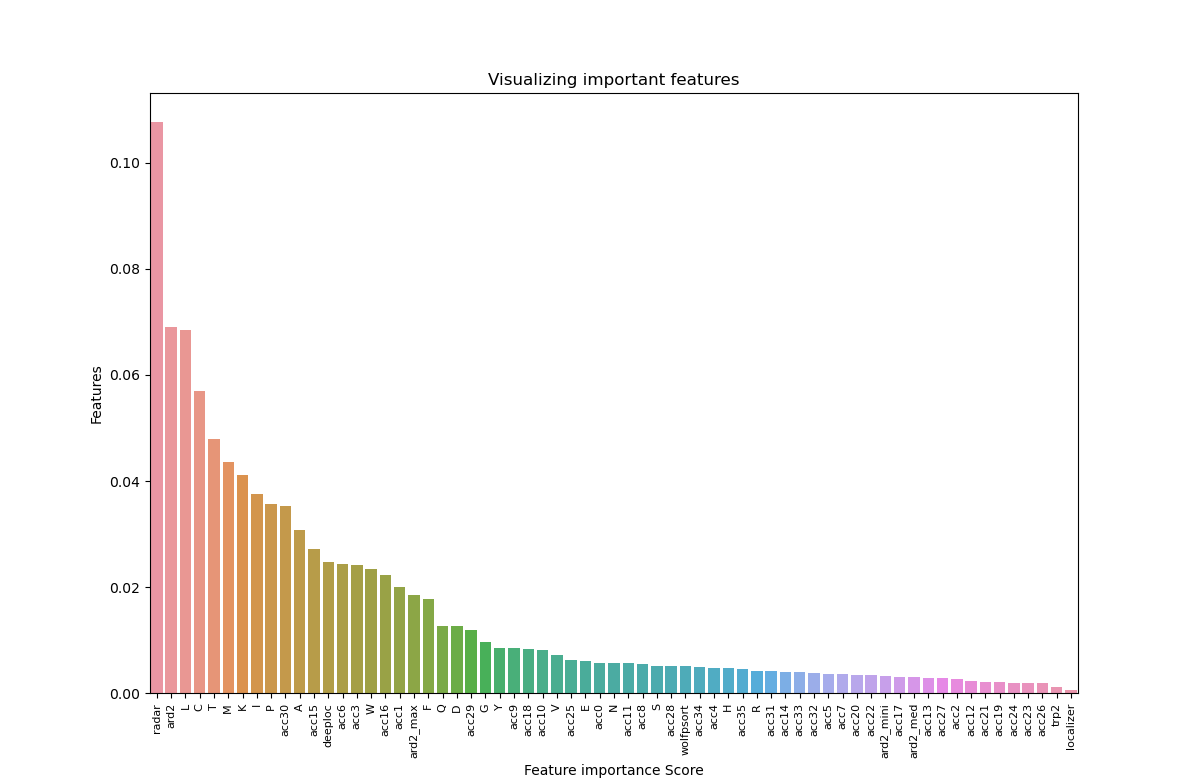
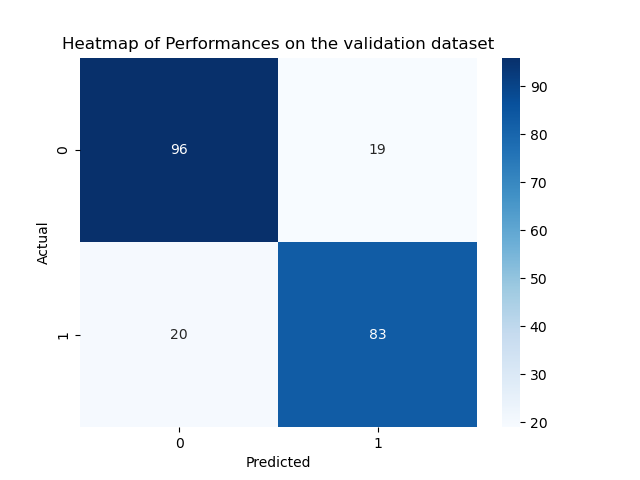
|  |  |  |
| --- | --- | --- |
|  | **Définition** | **Valeur (ou attribut)** |
| **criterion** | Fonction de mesure la qualité d’un échantillon  https://www.kaggle.com/questions-and-answers/196752 | « gini » (défault) |
| **n\_estimator** | Nombre d’arbres de la forêt | 500 |
| **min\_samples\_split** | Nombre minimum de données dans l’échantillon provenant de la dataset | 30 |
| **min\_samples\_leaf** | Nombre minimum d’échantillons à un nœud | 1 (défault) |
| **max\_features** | Nombre de descripteurs pris en compte par échantillons | « auto » (défault) |

https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d

À la suite de cela nous avons calculé les performances de notre modèle, visible ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Précision** | **Sensibilité** | **Spécificité** |
| Modèle | 0.93 | 0.94 | 0.87 |

**Table  :** Performances du modèle.



A

B

Faux négatifs

Faux positifs

Vrais négatifs

Vrais positifs

**Figure x :** (A) Heatmap des performances sur le jeu de validation. (B) Barplot de l’importance des descripteurs pour aider à la bonne prédiction du modèle.

Ainsi notre modèle a de bonnes performances avec un faible taux de mauvaises prédictions et une fiabilité dans les bonnes prédictions de 93% (figure xA). De plus à l’aide de l’attribut « .feature\_importances\_() » nous pouvons déterminer les descripteurs les plus importants pour le modèle, donc ceux qui sont les plus décisifs à classifier nos protéines en α-solénoïdes (figure xB). Nous avons filtré et gardé uniquement les descripteurs ayant une importance supérieure à 0.02. Nous remarquons alors que les logiciels les plus importants pour le modèle sont radar (répétitions de séquences), ard2 (hélices α) et deeploc (localisation cellulaire). Pour ce qui est de radar et ard2 cela n’est pas surprenant et corrobore avec ce que l’on connaît en termes de structure des ROGEs. Localizer et targetp2 sont placés en dernières positions ce qui nous fait dire que l’on pourrait se passer de ces logiciels à l’avenir. Il est aussi intéressant de noter que la proportion de certains acides aminés au sein des séquences est importante pour les prédictions : L (Leucine), C (Cytosine), T (Thréonine), M (Méthionine), K (Lysine), I (Isoleucine), P (Proline), A (Adénine), W (Tryptophane). Ces acides aminés sont pour la majorité apolaires (L, M, I, P, A, W). Une hypothèse probable est que cela pourrait s’expliquer que [Kappel] les protéines en α-solénoïdes sont constituées d’un cœur hydrophobe (dû à ses résidus apolaires qui le composent), au même endroit où vient se loger la molécule avec laquelle la protéine interagit. La détection d’un tel cœur hydrophobe par l’intermédiaire de la composition en acides aminés apolaires d’une protéine permettrait donc de mieux reconnaître nos protéines cibles.

Avec la figure xB des facteurs d’ACC en particuliers ont été soulignés. Un tableau des correspondances sera donné en annexe. Les ACC retenus par le modèle sont présentés ci-dessous : 🡪 mettre table des correspondance des acc en annexe

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Acc30** | **Acc15** | **Acc6** | **Acc3** | **Acc16** | **Acc1** |
| **Facteurs** | Z1-Z2 | Z2-Z1 | Z1-Z1 | Z1-Z1 | Z3-Z1 | Z2-Z2 |
| **Lag** | 4 | 1 | 3 | 2 | 1 | 1 |

**Table  :** Acc déterminants pour la prédiction du modèle.

Nous remarquons que la covariance au lag 1 était plus présente que celle des autres lag. De même pour la covariance entre même facteurs de Z-scales. Dans quasiment tous les cas de figures le premier Z-scales (Z1) est impliqué dans l’importance de l’ACC. Nous n’avons pas encore d’explications ou d’hypothèse mais il y est intéressant de le souligner.

Ensuite nous avons donc appliqué le protocole présenté plus haut sur les protéomes de *C. reinhardtii* et *A. Thaliana*. À la suite de cela nous obtenons une dataframe comme suivante : 🡪 surement annexes plutôt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Index** | **Radar** | **Ard2\_nombr\_**  **linker** | **Ard2\_proba\_**  **min** | **Ard2\_proba\_med** | **Ard2\_proba\_**  **max** |
| Identifiant  Protéique | Proportion de la séquence en répétition[[4]](#footnote-4) | Nombre de linker | Probabilité minimale d’un acide aminé d’être linker | Médiane des probabilités pour chaque acide aminé d’être linker | Probabilité maximale d’un acide aminé d’être linker |
| >NP\_001030614.1  (A. Thaliana) | 0.0 | 14.0 | 0.11 | 0.14 | 0.89 |

*(Suite de la dataframe)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Localizer** | **Deeploc** | **Targetp2[[5]](#footnote-5)** | **Acc0** | **Acc1** | **…** |
| 1 : adressé au chloroplaste  0 : autre | Probabilité maximale d’adressage entre les organites | 0.0 : NoTP  1.0 : cTP  2.0 : mTP  3.0 : iTP | Première valeur d’ACC sur Z-scales | Seconde valeur d’ACC sur Z-scales | … |
| 0.0 | 0.0075 | 0.0 | -0.344338870431894 | 0.453740531561462 | … |

*(Suite de la dataframe)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Acc35** | **A** | **S** | **W** | **…** | **X** |
| 36ième valeur d’ACC sur Z-scales | Fréquence d’apparition au sein de la séquence | Fréquence d’apparition au sein de la séquence | Fréquence d’apparition au sein de la séquence | Fréquence d’apparition au sein de la séquence | Fréquence d’apparition au sein de la séquence |
| ACC35 | 0.02761904761904762 | FS | FW | … | Fx |

**Table 3 :** Exemple pour une protéine de la dataframe finale obtenue. >NP\_001030614.1 est une protéine non prédite en alpha solénoïde par le modèle. Un exemple de résultat obtenu avec une protéine ensuite prédite comme ROGE sera présentée en annexe.

Nous avons alors pu déterminer au sein des deux protéomes quelles étaient les protéines α-solénoïdes. Ensuite pour ne garder que les protéines ROGEs (α-solénoïdes adressées aux organites) nous avons effectué un nouveau filtrage. Les résultats seront résumés dans la table ci-dessous.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A. Thaliana** | **C. reinhadtii** | **Total** |
| Nombre d’α-solénoïdes prédites | 1055 | 1716 | 2771 |
| Après filtrage (adressage deeploc) | 115 | 252 | 367 |
| Nouvelles protéines | 107 | 244 | 351 |
| Taille du protéome | 36699 | 15081 | 51780 |

**Table x :** Résultats du modèle de prédiction sur les organismes modèles A. thaliana et C. reinhardtii. On obtient in fine 351 nouvelles protéines ROGEs sur 51780 (soit ~0.29% du génome d’A. Thaliana et ~1.6% du génome de C. Reinhardtii).

Ensuite pour avoir une meilleure compréhension de nos résultats nous les avons comparé avec les jeux de donnés positifs et négatifs ainsi que ceux obtenus par la doctorante Céline Cattelin. Sa méthode repose sur des alignements et comparaisons de motifs (protocole disponible en annexe).

*Négatifs*

*Positifs*

B

A

107 OPR

390 PPR

12

545

*Comparaison et alignement de motifs*

*Machine Learning :*

*α-solénoïdes prédites*

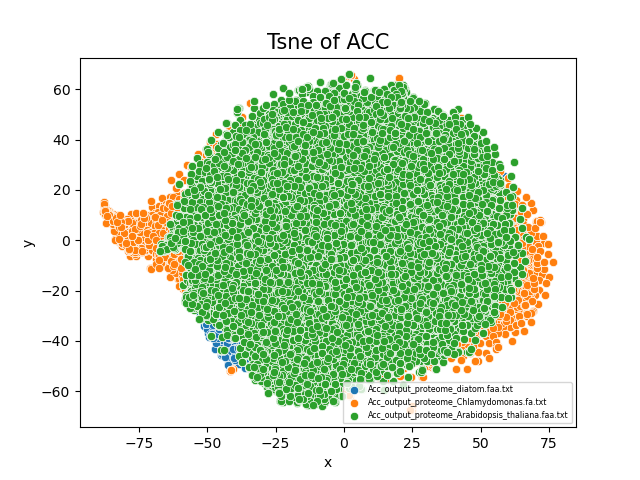
**Figure x :** Comparaison des résultats avec (A) les témoins positifs et négatifs (B) le travail de Céline Cattelin.

(C) après filtrage sur l’adressage.

Ces résultats ont montré que parmi les 2271 protéines candidates ROGEs il y avait déjà 545 protéines connues en α-solénoïde et 12 dont on sait qu’elles ne le sont pas. Cela correspond aux performances de notre modèle (93% de précision) donc nous pouvons penser que les résultats sont cohérents. Il s’agit donc d’investiguer sur les 1738 protéines restantes.

De plus nous avons comparé sur les 2271 protéines lesquelles étaient en OPR ou PPR. Nous trouvons donc 118 OPR et ? PPR, dont respectivement 107 et 390 ont déjà été trouvées par Céline Cattelin. 🡪 à voir après filtrage cb il en reste

Ensuite, pour avoir une idée globale de nos données nous avons effectué des t-sne sur les résultats obtenus. On cherche à pouvoir discriminer nos protéomes et déterminer s’il y a une différence significative entre eux. Si ce n’est pas le cas, alors nous pouvons continuer l’analyse.

**

B

A

**Figure 8 :** (A) t-sne sur les fréquences. Vert : Chlamydomonas, orange : Aradopsis Thalina, bleu : diatomées. (B) Tsne sur les ACC. Vert : Arabidopsis Thaliana, orange : Chlamydomonas, bleu : diatomées.

Chaque couleur correspond à un protéome et chaque point à une protéine de ce protéome.

On observe dans ces t-sne que chacune des trois espèces que les points déterminés par l’algorithme de la t-sne ont une forte tendance à se superposer pour les calculs des fréquences et les ACC. Cela signifie qu’il n’y a pas de différences notables entre la composition en acide aminé des différents protéomes. Donc les protéines qui constituent Arabidopsis Thaliana, Chlamydomonas reinhartii et les diatomées ont globalement les mêmes proportions d’acide aminé. Nous pouvons avoie la même raisonnement avec les ACC. En effet nous pouvons déduire de la figure 4B que les ACC de ces espèces sont sensiblement les mêmes. Nous en déduisons que les protéines de chaque protéomes sont équivalentes en terme de Z-scales et donc qu’elles ont les mêmes propriétés physico-chimiques. Étant donné que ces trois espèces sont proches les unes des autres, il est normal d’avoir ce type de résultats.

Avec les dataframes finales des échantillons positifs et négatifs nous avons de la même manière effectué une Tsne.



**Figure 9 :** Tsne comparative des échantillons positifs et négatifs, en prenant en compte tous les descripteurs obtenus.

Nous observons très clairement que les échantillons positifs et négatifs se distinguent sur la Tsne. On peut donc en conclure que ces échantillons ont bien bien des caractéristiques qui permettent de les différencier. Ainsi les descripteurs choisis sont considérés comme utile à notre étude et ces conclusions nous permettent de continuer l’analyse. Nous pouvons donc faire apprendre notre modèle sur ces protéomes.

Ss regroupements 🡪 class de prot ?

Colorer points en fct de l’espèce

Test de proportion sur le nb de linker et médiane de proba chez ard2 🡪 test significatif.

Insérer boxplots des outils et plus tard des aa importants selon le modèle

Discoussion

***ANNEXES***

Peptide d’adressage

3’

*Absence de domaine TM*

5’

1. Détection domaines TM

Protocole

*Présence de motifs répétés*

3’

68

5’

2. Détection motifs répétés

*Présence de paires d’hélice*

L

L

α

α

α

α

3’

68

5’

4. Prédiction localisation intracellulaire

3. Détection paires d’hélices

**Organites** (mitochondrie, chloroplaste)

**Annexe 1 :** Protocole mis en place pour identifier les protéines en alpha solénoïdes. Si une protéine passe tout ces filtre alors elle est candidate pour faire partie des protéines ROGEs. La présence d’hélice α est détectée par l’intermédiaire d’acides aminés « linker » (L) qui relient deux hélices entre elles.

|  |  |  |
| --- | --- | --- |
| **Outil** | **Prédiction** | **Information** |
| TMHMM | Structure transmembranaire | Identifiants des protéines sans domaine transmembranaire (après le 68e acide aminé) |
| Radar | Répétitions de séquences | Nombre de répétitions et proportion de la séquence qui possède des répétitions |
| Ard2 | Acide aminés linker d’hélice α | Nombre de linker, probabilités d’être linker |
| Targetp2 | Présence de pré-séquence en Nter (peptide d’adressage) | Adressage[[6]](#footnote-6) |
| Deeploc | Adressage | Probabilité d’adressage pour la mitochondrie et le chloroplaste |
| Wolfpsort | Adressage | Identifiants des protéines adressées à la mitochondrie ou au chloroplaste |
| Localizer | Adressage | Identifiants des protéines adressées au chloroplaste |

**Annexe 2 :** Tableau récapitulatif des outils utilisés et des informations recueillies après parsing des outputs.

**Annexe 3.1 :** Représentation du calcul des ACC pour un peptide donné au lag 4. Un peptide est caractérisé par r\*9 = 4\*9 = 36 valeurs d’ACC.



*Facteurs j et k, position i*

*Même facteur j, position i*

**Annexe 3.2 :** Formule du calcul des ACC. 🡪 annexe

|  |  |  |  |
| --- | --- | --- | --- |
| **Acc0** | Z1-Z1 lag1 | **Acc18** | Z1-Z2 lag2 |
| **Acc1** | Z2-Z2 lag1 | **Acc19** | Z1-Z3 lag2 |
| **Acc2** | Z3-Z3 lag1 | **Acc20** | Z2-Z3 lag2 |
| **Acc3** | Z1-Z1 lag2 | **Acc21** | Z2-Z1 lag2 |
| **Acc4** | Z2-Z2 lag2 | **Acc22** | Z3-Z1 lag2 |
| **Acc5** | Z3-Z3 lag2 | **Acc23** | Z3-Z2 lag2 |
| **Acc6** | Z1-Z1 lag3 | **Acc24** | Z1-Z2 lag3 |
| **Acc7** | Z2-Z2 lag3 | **Acc25** | Z1-Z3 lag3 |
| **Acc8** | Z3-Z3 lag3 | **Acc26** | Z2-Z3 lag3 |
| **Acc9** | Z1-Z1 lag4 | **Acc27** | Z2-Z1 lag3 |
| **Acc10** | Z2-Z2 lag4 | **Acc28** | Z3-Z1 lag3 |
| **Acc11** | Z3-Z3 lag4 | **Acc29** | Z3-Z2 lag3 |
| **Acc12** | Z1-Z2 lag1 | **Acc30** | Z1-Z2 lag4 |
| **Acc13** | Z1-Z3 lag1 | **Acc31** | Z1-Z3 lag4 |
| **Acc14** | Z2-Z3 lag1 | **Acc32** | Z2-Z3 lag4 |
| **Acc15** | Z2-Z1 lag1 | **Acc33** | Z2-Z1 lag4 |
| **Acc16** | Z3-Z1 lag1 | **Acc34** | Z3-Z1 lag4 |
| **Acc17** | Z3-Z2 lag1 | **Acc35** | Z3-Z2 lag4 |

**Annexe 4:** Table de correspondance des ACC

Mettres des exemple d’output des logiciels + dataframe finale obtenues sur les pos et négatifs

***RÉFÉRENCES***

[] Guo & al. (2022) « A computational method for predictiong nucleocapsid protein in retroviruses », *Scientific Reports, Volume 12, page 524, 10.1038/s41598-021-03182-2.*

[] Sandaruwan & al. (2021) « An improved deep learning model for hierarchical classification of protein families », *PloS ONE, Volume 16, Numéro 10, 10.1371/journal.pone.0258625*.

[] Takenaka & al. (2021) « DYW domain structures imply an unusual regulation principle in plant organellar RNA editing catalysis », *Nature Catalysis, Volume 4, Numéro 6, pages 510-522, 10.1038/s41929-021-00633-x.*

[] Zhao & al. (2021) « DescribePROT : database of amino acid-level protein structure and function predictions », *Nucleic Ancids Reasearch, Volume 49, Numéro D1, pages D298-D308, 10.1093/nar/gkaa931.*

[] Chatzimparm & al. (2020) « t-viSNE : Interactive Assessment and Interpretation of t-SNE Projections », *IEE Trans. Visual. Comput. Graphics, Volume 26, Numéro 8, pages 2696-2714, 10.1109/TVCG.2020.2986996.*

[] Garrido & al. (2020) « Evidence Supporting an Animicrobial Origin of Targeting Peptides to Endosymbiotic Organelles », *Cells, Volume 9, Numéro 8, pages 1795, 10.3390/cells9081795.*

[] Falciatore & al. (2020) « Diatom Molecular Research Comes of Age : Model Species for Studying Phytoplaktion Biology and Diversity », *The Plant Cell, Volume 32, Numéro 3, pages 547-572, 10.1105/tpc.19.00158.*

[] Gutmann & al. (2020) « The Expansion and Diversification of Pentratricopeptide Repeat RNA-Editing Factors in Plants », *Molecular Plant, Volume 13, Numéro 2, pages 215-230, 10.1016/j.molp.2019.11.002.*

[] Burki & al. (2020) «  The new Tree of Eukaryotes », *Trends in Ecology and Evolution, Volume 35, Numéro 1, pages 43-55, 10.1016/j.tree.2019.08.008.*

[] Nanni & Brahman (2020) « Set of Approaches Based on Position Specific Scoring Matrix and Amino Acid Sequence for Primary Category Enzyme Classification », *Journal of Artificial Intelligence and Systems, Volume 2, Numéro 1, pages 38-52, 10.33969/AIS.2020.21004.*

[] Ponce-Toledo, Lopez-Garcia & Moreira (2019) « Horizontal and endosymbiotic gene transfer in early plastid evolution », *New Phytologist, Volume 224, Numéro 2, pages 618-624, 10.1111/nph.15965.*

[] Lv & al. (2019) «  A random Forest Sub-Golgi Protein Classifier Optimized via Dipeptide and Amino Acid Composition Features », *Frontiers in Bioengineering and Biotechnology, Volume 7, pages 215, 10.3389/fbioe.2019.00215*.

[] Graham & al (2019) « A Random Forest Sub-Golgi Protein Classifier Optimized via Dipeptide and Amino Acid Composition Features », *Critical Reviews in Biochemistry and Molecular Biology, Volume 54, Numéro 2, pages 103-118, 10.1080/10409238.2019.1590305.*

[] Luttrel & al. (2019) « Predicting protein residue-residue contacts using random forests and deep networks », *BMC Bioinformatics, Volume 20, Numéro S2, pages 100, 10.1186/s12859-019-2627-6.*

[] Hakala & al. (2019) « Neural network and random forest models in protein function prediction », *bioRxiv, pages 18.*

[] Hillebrand & al. (2018) « Identification of clustered organellar short (cos) RNAs and of a conserved family of organellar RNA-binding proteins, the heptatricopeptide repeat proteins, in the malaria parasite », *Nucleic Acids Research, 10.1093/nar/gky710.*

[] Seo & al. (2018) « DeepFam: deep learning based alignment-free method for protein family modeling and prediction », *Bioinformatics , Volume 34, Numéro 13, pages i254-i262, 10.1093/bioinformatics/bty275.*

[] Taherzadeh & al. (2018) « Structure-based prediction of protein– peptide binding regions using Random Forest », *Bioinformatics, Volume 34, Numéro 3, pages 477-484, 10.1093/bioinformatics/btx614.*

[] Kathuria & al. (2018) « Predicting the protein structure using random forest approach », *Procedia Computer Science, Volume 132, pages 1654-1662, 10.1016/j.procs.2018.05.134.*

[] Schietgat & al. (2018) « A machine learning based framework to identify and classify long terminal repeat retrotransposons », *PLOS Computational Biology, Volume 14, Numéro 4, pages e1006097, 10.1371/journal.pcbi.1006097.*

[] Sarica & al. (2017) « Random Forest Algorithm for the Classification of Neuroimaging Data in Alzheimer's Disease: A Systematic Review », *Frontiers in Aging Neuroscience, Volume 9, pages 329, 10.3389/fnagi.2017.00329.*

[] Zielinski & al. (2017) « Deep learning approach to bacterial colony classification », *PLOS ONE, Volume 12, Numéro 9, 10.1371/journal.pone.0184554.*

[] Fučíková & al. (2016) « Chloroplast phylogenomic data from the green algal order Sphaeropleales (Chlorophyceae, Chlorophyta) reveal complex patterns of sequence evolution », *Molecular Phylogenetics and Evolution, Volume 98, pages 176-183, 10.1016/j.ympev.2016.01.022.*

[] Turesson & al. (2016) « Machine Learning Algorithms for Automatic Classification of Marmoset Vocalizations », *PLOS ONE, Volume 11, Numéro 9, 10.1371/journal.pone.0163041.*

[] Marx, Wunsch & Kück (2015) « The Octatricopeptide Repeat Protein Raa8 Is Required for Chloroplast *trans* Splicing », *Eukaryotic Cell, Volume 14, Numéro 10, pages 998-1005, 10.1128/EC.00096-15.*

[] Libbrecht & Noble (2015) « Machine learning applications in genetics and genomics », *Nature Reviews Genetics, Volume 16, Numéro 6, pages 321-332, 10.1038/nrg3920.*

[] Boulouis & al. (2015) « Spontaneous Dominant Mutations in Chlamydomonas Highlight Ongoing Evolution by Gene Diversification », *The Plant Cell, Volume 27, Numéro 4, pages 984-1001, 10.1105/tpc.15.00010.*

[] Mohan & al. (2014) « Automatic classification of protein structures using physicochemical parameters », *Interdisciplinary Sciences: Computational Life Sciences, Volume 6, Numro 3, pages 176-186, 10.1007/s12539-013-0199-0.*

[] Barkan & Small (2014) « Pentatricopeptide Repeat Proteins in Plants », *Annual Review of Plant Biology, Volume 65, Numéro 1, pages 415-442, 10.1146/annurev-arplant-050213-040159.*

[] Ge & al. (2014) « Import Determinants of Organelle-Specific and Dual Targeting Peptides of Mitochondria and Chloroplasts in Arabidopsis thaliana », *Molecular Plant, Volume 7, Numéro 1, 10.1093/mp/sst148.*

[] Zhao & al. (2014) « Determining Effects of Non-synonymous SNPs on Protein-Protein Interactions using Supervised and Semi-supervised Learning », *PLoS Computational Biology, Volume 10, Numéro 5, pages e1003592, 10.1371/journal.pcbi.1003592.*

[] Jahandideh & al. (2014) « Improving the chances of successful protein structure determination with a random forest classifier », *Acta Crystallographica Section D Biological Crystallography, Volume 70, Numéro 3, pages 627-635, 10.1107/S1399004713032070.*

[] Fournier & al. (2013) « Functional and Genomic Analyses of Alpha-Solenoid Proteins », *PLOS ONE, Volume 8, Numéro 11, 10.1371/journal.pone.0079894.*

[] Tourasse & al. (2013) «  PPR proteins of green algae », *RNA Biology, Volume 10, Numéro 9, pages 1526-1542, 10.4161/rna.26127.*

[] He & al. (2012) « Modeling the QSAR of ACE-Inhibitory Peptides with ANN and Its Applied Illustration », *International Journal of Peptides, Volume 2012, pages 1-9, 10.1155/2012/620609.*

[] Leliaert & al. (2012) « Phylogeny and Molecular Evolution of the Green Algae », *Critical Reviews in Plant Sciences, Volume 31, Numéro 1, pages 1-46, 10.1080/07352689.2011.615705.*

[] Fukui & Kuramitsu (2011) « Structure and Function of the Small MutS-Related Domain », *Molecular Biology International, Volume 2011, pages 1-9, 10.4061/2011/691735.*

[] Lin & Chen (2011) « Prediction of thermophilic proteins using feature selection technique », *Journal of Microbiological Methods, Volume 84, Numéro 1, pages 67-70, 10.1016/j.mimet.2010.10.013.*

[] Jain & Hirst (2010) « Automatic structure classification of small proteins using random forest », *BMC Bioinformatics, Volume 11, Numéro 1, pages 364, 10.1186/1471-2105-11-364.*

[] Madera & al. (2010) « Improving protein secondary structure prediction using a simple k-mer model », *Bioinformatics, Volume 26, Numéro 5, pages 596-602, 10.1093/bioinformatics/btq020.*

[] Wold & al. (1993) « DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures », *Analytica Chimica Acta, Volume 277, Numéro 2, pages 239-253, 10.1016/0003-2670(93)80437-P.*

[] Hellberg & al (1987) « Peptide quantitative structure-activity relationships, a multivariate approach », *Journal of Medicinal Chemistry, Volume 30, Numéro 7, 10.1021/jm00390a003.*

1. Chaque élément à i+1 ne dépend que de la valeur à la position i, les valeurs antérieurs ne permettent aucune ou qu’une très faible prédiction [↑](#footnote-ref-1)
2. Noyau, Cytoplasme, extracellulaire, mitochondriale, membranaire, reticulum endoplasmique, chloroplaste, appareil de Golgi, Lysosome/vacuole et peroxysome. [↑](#footnote-ref-2)
3. Sorties possibles : SP = signal peptide, mTP = mitochondrial transit peptide, cTP = chloroplast transit peptide, iTP = thylakoid transit peptide, NoTP = aucun signal [↑](#footnote-ref-3)
4. Pour des répétitions de longueur comprises entre 29 et 46. En effet ce sont des types de répétitions caractéristiques des OPR, TPR et PPR. [↑](#footnote-ref-4)
5. Sorties possibles : SP = signal peptide, mTP = mitochondrial transit peptide, cTP = chloroplast transit peptide, iTP = thylakoid transit peptide, NoTP = aucun signal [↑](#footnote-ref-5)
6. Sorties possibles : SP = signal peptide, mTP = mitochondrial transit peptide, cTP = chloroplast transit peptide, iTP = thylakoid transit peptide, NoTP = aucun signal [↑](#footnote-ref-6)