Titre: Title:	Modèle de prédiction de la phosphorylation d'acides aminés pour l'identification de biomarqueurs associés aux maladies neurodégénératives
Auteur: Author:	Uriel Manseau-Pérez
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Manseau-Pérez, U. (2025). Modèle de prédiction de la phosphorylation d'acides aminés pour l'identification de biomarqueurs associés aux maladies neurodégénératives [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/68170/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Modèle de prédiction de la phosphorylation d'acides aminés pour l'identification de biomarqueurs associés aux maladies neurodégénératives

URIEL MANSEAU-PÉREZ
Département de mathématiques et génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques

Août 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Modèle de prédiction de la phosphorylation d'acides aminés pour l'identification de biomarqueurs associés aux maladies neurodégénératives

présenté par **Uriel MANSEAU-PÉREZ**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Nadia LAHRICHI, présidente
Richard LABIB, membre et directeur de recherche
Luc ADJENGUE, membre

DÉDICACE

À mon père,
dont le regard continue de me dire ce que les mots ne peuvent plus . . .
Ces lignes informes ne sont que ma modeste réponse à ton silence.

Et à tous ceux qui savent encore trouver promesse
dans ce qui ne se dit plus,
gardez l'espoir.

REMERCIEMENTS

Je souhaite tout d'abord exprimer ma profonde reconnaissance à mon directeur de recherche, Richard Labib, pour m'avoir ouvert les portes du monde de la recherche, pour la richesse de son savoir partagé, la rigueur de son encadrement et la confiance qu'il m'a témoignée tout au long de ce parcours.

Je tiens également à remercier Francesca Cicchetti et Melanie Alpaugh, qui nous ont offert l'opportunité de contribuer à leurs travaux de recherche d'une portée scientifique et humaine essentielle.

J'aimerais aussi remercier mes amis, qui sauront se reconnaître, dont la présence, les encouragements et l'humour ont été des ancrages précieux dans les moments d'incertitude.

Enfin, je souhaite exprimer toute ma gratitude à ma mère, Leticia, et à ma sœur, Maya, pour leur soutien indéfectible tout au long de cette aventure.

RÉSUMÉ

À l'heure actuelle, le diagnostic et le suivi de la progression de nombreuses maladies neurodégénératives représentent un défi médical majeur, largement attribuable à l'absence de biomarqueurs fiables. L'identification de nouveaux biomarqueurs repose majoritairement sur de longues et coûteuses expériences en laboratoire. Ces expériences, qui visent à quantifier la phosphorylation des acides aminés au sein des protéines, pourraient être optimisées. Les approches computationnelles offrent une solution prometteuse en permettant de cibler en amont les acides aminés les plus susceptibles d'être phosphorylés, guidant ainsi plus efficacement la recherche expérimentale et réduisant les ressources consacrées à l'étude de biomarqueurs à faible potentiel. Dans ce contexte, l'objectif principal de ce mémoire est de développer un modèle de prédiction de la phosphorylation d'acides aminés à la fois plus performant que les approches existantes et offrant un meilleur niveau d'interprétabilité, un critère fondamental pour son adoption par la communauté scientifique.

Pour atteindre notre objectif, nous avons adopté une approche systématique d'apprentissage machine, au sein de laquelle nous avons intégré trois contributions originales visant à surmonter les limitations des modèles actuels. Premièrement, nous avons développé une méthode de modélisation tridimensionnelle robuste qui caractérise plus finement l'environnement spatial des sites de phosphorylation. Deuxièmement, nous avons conçu une stratégie d'échantillonnage novatrice pour adresser à la fois le fort déséquilibre des classes et l'incertitude inhérente aux données disponibles. Finalement, nous avons mis au point une méthode de sélection de variables qui explore efficacement l'espace des caractéristiques pour identifier les combinaisons les plus performantes, tout en conservant une interprétabilité biologique.

L'intégration de ces contributions a mené à la création d'un modèle final dont les performances surpassent celles des modèles de référence recensés dans la littérature. Pour valider notre approche sur un cas concret, nous avons appliqué le modèle à l'identification de biomarqueurs potentiels pour la maladie de Huntington, une maladie neurodégénérative pour laquelle notre participation à une étude scientifique connexe a motivé notre choix. Cette application a permis de générer une liste hiérarchisée d'acides aminés de la protéine tau, offrant des cibles prioritaires pour de futures investigations expérimentales et illustrant le potentiel de notre outil pour accélérer la découverte de biomarqueurs.

ABSTRACT

At present, diagnosing and monitoring the progression of many neurodegenerative diseases represents a major medical challenge, largely due to the lack of reliable biomarkers. The identification of new biomarkers relies mainly on long and costly laboratory experiments. These experiments, which aim to quantify the phosphorylation of amino acids within proteins, could be optimized. Computational approaches offer a promising solution by enabling upstream targeting of the amino acids most likely to be phosphorylated, thus guiding experimental research more effectively and reducing the resources devoted to studying biomarkers with low potential. In this context, the main aim of this thesis is to develop a phosphorylation prediction model that not only outperforms existing approaches, but also offers a better level of interpretability, a fundamental criterion for its adoption by the scientific community.

To achieve our goal, we have adopted a systematic machine learning approach, within which we have integrated three original contributions aimed at overcoming the limitations of current models. Firstly, we developed a robust three-dimensional modeling method that more finely characterizes the spatial environment of phosphorylation sites. Secondly, we designed an innovative sampling strategy to address both the strong class imbalance and the uncertainty inherent in the available data. Finally, we developed a variable selection method that intelligently explores the feature space to identify the best performing combinations, while retaining biological interpretability.

The integration of these contributions has led to the creation of a final model whose performance surpasses that of reference models in the literature. To demonstrate the practical usefulness of our approach, we applied this model to the identification of potential biomarkers for Huntington's disease, a neurodegenerative disorder for which our participation in a related scientific study motivated this choice. This application generated a hierarchical list of amino acids in the tau protein, offering priority targets for future experimental investigations and illustrating the potential of our tool to accelerate biomarker discovery.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AIC	Critère d'information d'Akaike
ANOVA	Analyse de la variance
AUC	« Area under the curve »
BCA	« Bias-Corrected and accelerated »
BIC	Critère d'information bayésien
CNN	« Convolutional Neural Network »
DM	Distance de Mahalanobis
FN	« False Negative »
FP	« False Positive »
IQR	« Interquartile Range »
KDC	« Cyclin-dependent kinases »
LDA	« Linear Discriminant Analysis »
LSTM	« Long Short-Term Memory »
mRMR	« Minimum Redundancy Maximum Relevance »
MVS	Machine à vecteurs de support
PELM	Phospho.ELM
PR-AUC	« Precision-Recall Area Under the Curve »
RNA	Réseau de neurones artificiels
ROC-AUC	« Receiver Operating Characteristic »
SEQ	Fenêtre séquentielle
SLA	Sclérose latérale amyotrophique
SPT	Surface polaire topologique
TN	« True Negative »
TP	« True Positive »
VC	Validation croisée
XGBoost	« Extreme Gradient Boosting »

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Problématique des tauopathies

Les tauopathies sont une grande famille de maladies neurodégénératives qui comprennent, mais sans s'y limiter, la maladie d'Alzheimer, certains types de parkinsonisme, la sclérose latérale amyotrophique (SLA) et plusieurs types de démence. Ces maladies partagent comme caractéristique principale l'accumulation excessive, à l'intérieur des neurones du cerveau, d'une protéine spécifique appelée tau. Une protéine est une longue chaîne composée d'acides aminés. Une quantité trop élevée de la protéine tau mène à la formation d'agrégats toxiques pour les neurones, provoquant progressivement leur dégénérescence.

Les tauopathies affectent des millions de personnes dans le monde chaque année. Par exemple, la maladie d'Alzheimer touche environ 55 millions de personnes et 10 millions de nouveaux cas sont diagnostiqués annuellement [1]. La maladie de Parkinson affecte environ 10 millions de personnes à travers le monde, avec plusieurs centaines de milliers de nouveaux cas chaque année [2]. En plus des conséquences sur les personnes touchées et leur entourage, ces maladies pèsent lourd sur les systèmes de santé. À titre d'exemple, au Canada seulement, on estime à plus de 40 milliards les coûts associés à la maladie d'Alzheimer [3].

Malheureusement, aucun traitement ne permet actuellement d'inverser complètement les symptômes ou d'en arrêter la progression. Même qu'à ce jour, poser un diagnostic fiable pour les tauopathies reste un défi médical important. Cela s'explique par le fait que la plupart de ces maladies ne disposent toujours pas d'indicateurs biologiques mesurables permettant de poser un diagnostic ou de suivre la progression des symptômes. Ces indicateurs sont appelés des biomarqueurs. Un biomarqueur est dit mesurable si sa quantité peut être précisément déterminée par des analyses sanguines, d'imagerie médicale ou d'autres tissus.

Pour certaines tauopathies, comme la maladie d'Alzheimer, la recherche de biomarqueurs montre des résultats prometteurs [1] [2] [4]. En effet, les études montrent que la phosphorylation, une réaction chimique impliquant l'ajout d'un groupe phosphate, de certains acides aminés de la protéine tau pourrait servir de biomarqueur. Ainsi, un niveau trop élevé de phosphorylation d'acides aminés spécifiques de la protéine tau causerait la formation d'agrégats toxiques au sein des neurones. Dans le cadre de ce mémoire, nous développerons un modèle de prédiction de la phosphorylation d'acides aminés et utiliserons ce modèle sur la protéine tau afin d'identifier de potentiels biomarqueurs pour une maladie neurodégénérative spécifique, la maladie de Huntington. Cette pathologie héréditaire se caractérise par des mou-

vements involontaires, un déclin cognitif et des troubles psychiatriques. Également appelée chorée de Huntington, cette maladie est classifiée comme une tauopathie secondaire. Cela signifie qu'il n'est pas encore établi si l'accumulation toxique de la protéine tau est la cause de la dégénérescence neuronale ou si elle en est plutôt une conséquence [5] [6].

1.2 Enjeux et objectifs du mémoire

La chaîne protéique de tau est constituée de 758 acides aminés. Il existe 20 types d'acides aminés et toutes les protéines du corps humain, dont tau, sont formées d'une combinaison unique de ceux-ci. La phosphorylation ne peut se produire que sur les 3 types d'acides aminés suivants : la sérine, la thréonine et la tyrosine. Cela est dû à la nature de leur chaîne latérale, une composante de leur structure moléculaire, qui permet à la phosphorylation de se produire.

Au sein de la protéine tau, on y retrouve 79 sérines, 50 thréonines et 6 tyrosines, pour un total de 135 acides aminés, également appelés résidus, sur lesquels la phosphorylation est possible. Toutefois, à ce jour, la phosphorylation n'a été confirmée expérimentalement que sur 26 sérines, 13 thréonines et 4 tyrosines distinctes parmi ces 135 résidus [7]. Cette différence s'explique par les limites techniques des méthodes disponibles pour détecter la phosphorylation, qui ne permettent pas d'identifier systématiquement celle-ci sur tous les acides aminés au sein des protéines. La validation expérimentale de la phosphorylation s'effectue principalement à l'aide de deux techniques : la spectrométrie de masse, une méthode qui mesure la masse moléculaire des fragments de protéines afin de localiser les acides aminés phosphorylés, et le buvardage de Western, qui utilise des anticorps spécifiques pour détecter les acides aminés phosphorylés. En fonction du type d'acide aminé étudié, de sa position au sein de la protéine, ainsi que de la protéine elle-même, l'approche expérimentale doit être soigneusement adaptée. Valider expérimentalement de façon exhaustive tous les événements plausibles de phosphorylation au sein des protéines du corps humain est un défi colossal, voire impossible. En effet, une protéine contient en moyenne 430 acides aminés [8] et on estime qu'il existe environ 20 000 gènes dans le corps humain, chacun pouvant coder une protéine distincte [9].

Devant les limitations des méthodes expérimentales, des approches computationnelles sont développées afin de prédire la probabilité qu'un acide aminé soit le site d'une phosphorylation. Ces méthodes exploitent des bases de données contenant des sites de phosphorylation déjà identifiés expérimentalement, ainsi que des techniques d'apprentissage machine, afin d'estimer la probabilité de phosphorylation d'un acide aminé donné. Ces outils offrent aux chercheurs un moyen d'identifier en amont d'une analyse en laboratoire les acides aminés les plus susceptibles d'être phosphorylés, en vue de découvrir de nouveaux biomarqueurs. Cependant, les modèles actuellement disponibles n'offrent pas encore un niveau de performance

et d'interprétabilité suffisamment élevé pour permettre leur utilisation en toute confiance par la communauté scientifique [10]. Le premier objectif de ce mémoire est donc de développer un modèle plus précis de prédiction de la phosphorylation et avec un meilleur niveau d'interprétabilité.

Pour développer notre modèle, nous adoptons une approche systématique d'apprentissage machine qui intègre les étapes de modélisation des intrants, prétraitement des données, échantillonnage et équilibrage des classes ainsi que de sélection de variables. La performance de notre modèle sera comparée avec les modèles de la littérature. Le premier objectif de ce mémoire est divisé en 3 sous-objectifs, chacun étant directement lié à une contribution originale s'inscrivant dans une étape précise de l'approche systématique d'apprentissage machine.

Nos trois contributions répondent à des limitations identifiées au sein de différentes étapes de conception des modèles de prédiction de la phosphorylation dans la littérature. La première contribution porte sur la méthodologie de modélisation des sites de phosphorylation. Lors de la modélisation des sites de phosphorylation, nous avons constaté que plusieurs modèles ignorent l'aspect tridimensionnel des protéines, et ceux qui l'intègrent présentent certaines limitations que nous souhaitons pallier. Le premier sous-objectif est donc de développer une méthode de modélisation tridimensionnelle plus robuste. Nous montrons que le fait d'intégrer l'aspect tridimensionnel de manière rigoureuse permet d'augmenter les performances de prédiction.

La deuxième contribution concerne la méthode d'échantillonnage des acides aminés dans les jeux de données utilisés pour entraîner les modèles prédictifs. En effet, les jeux de données sur la phosphorylation présentent trois problèmes. Premièrement, on constate un déséquilibre important entre le nombre d'acides aminés phosphorylés et non phosphorylés. Deuxièmement, on observe une redondance au sein des données. Troisièmement, il existe de l'incertitude quant à savoir si certains acides aminés classés comme non phosphorylés le sont véritablement. La majorité des modèles de la littérature utilisent des approches d'échantillonnage similaires, qui n'adressent pas directement les problèmes cités précédemment. Dans ce contexte, notre deuxième sous-objectif est de développer une nouvelle méthode d'échantillonnage qui adresse directement les 3 problèmes dont souffrent les ensembles de données sur la phosphorylation.

Le troisième sous-objectif est lié aux méthodes de sélection de variables employées dans les modèles. Dû à la complexité des systèmes biologiques qui régissent la phosphorylation, il existe un grand nombre de variables pouvant être incluses dans les modèles et les interactions complexes possibles entre elles peuvent engendrer une explosion combinatoire des facteurs à considérer. De plus, puisqu'il s'agit ici d'un contexte médical, la nécessité de maintenir l'interprétabilité biologique des résultats limite le type et l'étendue des transformations pou-

vant être appliquées aux données. Afin de surmonter le défi associé à la forte dimensionnalité tout en préservant l'interprétabilité des résultats, nous proposons une méthode novatrice de sélection de variables qui permet non seulement de contrôler la réduction du nombre de variables incluses dans le modèle, mais aussi de déterminer un sous-ensemble de caractéristiques optimisant les performances.

CHAPITRE 2 REVUE DE LITTÉRATURE

La revue de littérature traite des méthodes employées par les modèles de prédiction de la phosphorylation d'acides aminés. Plus précisément, les méthodes de modélisation, d'échantillonnage et de sélection de variables sont revues.

2.1 Prédiction de la phosphorylation d'acides aminés

La phosphorylation est catalysée par des enzymes appelées kinases et ce concept est important car il sépare les modèles abordés dans cette section en 2 types : les modèles généraux et les modèles spécifiques. Les modèles généraux prédisent la probabilité de phosphorylation sans tenir compte de l'identité de la kinase responsable. À l'inverse, les modèles spécifiques prédisent la probabilité qu'un acide aminé soit phosphorylé par une kinase précise. On retrouve aussi quelques modèles hybrides, qui peuvent être utilisés en intégrant ou non l'identité spécifique de la kinase impliquée. Cette segmentation est importante pour notre revue de littérature car le type de modèle considéré a une influence sur les choix des méthodes de modélisation, d'échantillonnage et de sélection de variables, les domaines sur lesquels portent nos 3 contributions. Nous avons donc structuré cette section en 3 parties, chacune étant directement liée à l'un de nos 3 sous-objectifs. D'abord, nous présentons les méthodes utilisées pour modéliser les sites de phosphorylation. Ensuite, nous abordons les stratégies d'échantillonnage employées dans la littérature. Enfin, nous passons en revue les approches de sélection de variables.

2.1.1 Méthodes de modélisation des sites de phosphorylation

Pour prédire la probabilité qu'un résidu de sérine, de thréonine ou de tyrosine soit phosphorylé, les modèles de la littérature utilisent diverses stratégies pour modéliser l'information issue de l'environnement local. Dans notre contexte, l'environnement local correspond aux caractéristiques des acides aminés à proximité ainsi qu'aux propriétés structurales de la région protéique concernée. Afin de structurer cette sous-section, nous avons classé les approches de modélisation en trois catégories distinctes : séquentielle, structuro-séquentielle et tridimensionnelle.

Modélisation séquentielle

Les modèles séquentiels [11–23] modélisent l'environnement local à l'aide de fenêtres composées d'un certain nombre d'acides aminés. Ces fenêtres sont centrées sur le résidu pour lequel on prédit la probabilité de phosphorylation. Nous subdivisons les modèles séquentiels en 2 sous-groupes : les modèles séquentiels classiques et les modèles séquentiels avec jetons. Les modèles séquentiels classiques intègrent explicitement dans leur modélisation les caractéristiques des acides aminés présents dans la fenêtre. À l'inverse, les modèles séquentiels avec jetons ne fournissent aucune information relative aux acides aminés adjacents.

Hai Dang et al. (2008) [11], avec leur modèle CRPhos, proposent une modélisation séquentielle classique. Les auteurs utilisent une taille de fenêtre de 9 acides aminés, considérant donc les 4 acides aminés à gauche et les 4 acides aminés à droite du résidu d'intérêt. Afin de caractériser chaque fenêtre, les auteurs regroupent les 20 types d'acides aminés existants en 8 classes. Ces classes regroupent les résidus ayant des propriétés similaires. Ces propriétés physico-chimiques influencent directement les réactions chimiques susceptibles de se produire dans l'environnement local, en faisant une information pertinente. En effet, nous constatons que les propriétés physico-chimiques sont utilisées par la majorité des modèles de la littérature. La charge (positive, neutre, négative), la polarité (polaire, non polaire), le profil hydrophile/hydrophobe (tendance à interagir avec l'eau) et la solubilité en sont quelques exemples. La taille de fenêtre qu'utilisent Hai Dang et al. (2008) est unique, alors que d'autres, comme Luo et al. (2019) [23], emploient plusieurs tailles de fenêtres dans leur approche. En effet, les auteurs utilisent trois fenêtres de tailles différentes afin de mieux modéliser l'environnement local des sites de phosphorylation. Cette méthode permet d'intégrer des acides aminés potentiellement pertinents pour la prédiction, qui risqueraient autrement d'être exclus en utilisant une seule fenêtre de taille fixe. Le modèle de Hai Dang et al. (2008) semble mieux performer que le modèle de Luo et al. (2019), DeepPhos, qui utilise plusieurs tailles de fenêtres. Cependant, ces deux modèles sont difficilement comparables car CRPhos est spécifique aux kinases, alors que DeepPhos est un modèle général. Un modèle général est plus intéressant si l'accent est mis sur la détection des événements de phosphorylation, ce qui est notre cas, plutôt que sur la visibilité sur la kinase responsable.

Wang et al. (2022) [19] proposent une approche séquentielle avec jetons qui traite plutôt les séquences d'acides aminés comme des chaînes de caractères et font usage d'une technique d'encodage, la « tokenization ». Le modèle « apprend » une représentation latente des séquences lors d'une phase de pré-entraînement. Durant cette phase, le mécanisme d'attention [24] est utilisé et permet d'exploiter des tailles de fenêtres beaucoup plus grandes que les méthodes séquentielles classiques. En effet, le mécanisme d'attention est conçu pour avoir

une bonne capacité à capturer des relations entre des éléments éloignés dans une séquence de données. À titre d'exemple, les fenêtres utilisées par Wang et al. (2022) considèrent les 50 acides aminés adjacents aux sites de phosphorylation. Cet avantage a permis aux auteurs de développer un modèle avec des performances qui surpassent légèrement les modèles avec une approche séquentielle classique. Cependant, le processus de « tokenization », indispensable au mécanisme d'attention, transforme les entrées de sorte qu'il est difficile de comprendre ce qui se passe à l'intérieur de ces modèles. De plus, les représentations latentes « apprises » lors de l'étape de pré-entraînement ne sont pas directement interprétables, faisant en sorte qu'il est difficile de justifier biologiquement les résultats de prédiction. Nous verrons qu'il est possible d'adopter un style de modélisation classique, contribuant à conserver l'interprétabilité de notre modèle, tout en offrant une bonne performance.

Modélisation structuro-séquentielle

Les modèles suivant cette approche utilisent le concept de fenêtre, mais intègrent également des informations sur la structure locale des protéines. Dans la littérature, nous avons identifié trois caractéristiques qui sont utilisées pour décrire cette structure locale. Premièrement, le score de désordre, employé par Qiu et al. (2016) [25] et Dou et al. (2014) [26], représente la probabilité qu'un résidu se trouve dans une région intrinsèquement désordonnée de la protéine. Les auteurs utilisent l'outil DISOPRED [27] afin d'estimer ce score pour le résidu central d'une fenêtre.

Deuxièmement, la surface accessible au solvant, exploitée également par Dou et al. (2014), quantifie la portion de la surface du résidu pouvant être exposée à un solvant, l'eau dans notre cas. Cette mesure est pertinente pour prédire la phosphorylation, car elle permet de quantifier l'accessibilité des sites pour les kinases. Toutefois, Kumar et al. (2015) [28] ont montré que certains événements de phosphorylation peuvent se produire même avec une faible surface accessible, indiquant que d'autres aspects de la structure locale sont également à considérer.

Troisièmement, la structure secondaire est une caractéristique indiquant la présence de motifs précis fréquemment observés dans les protéines, tels que les hélices ou les feuillets. Dou et al. (2014) utilisent l'outil PSIPRED [29] pour obtenir la probabilité qu'un résidu central d'une fenêtre se trouve dans l'un de ces motifs structuraux, et incluent cette information supplémentaire dans leur modélisation.

Enfin, les travaux de Dou et al. (2014) et Qiu et al. (2016) montrent que les modèles intégrant des informations structurelles offrent une meilleure performance prédictive comparativement à des approches strictement séquentielles telles que Musite [21] ou PPRED [14]. Ces résultats suggèrent donc qu'utiliser des données structurelles en complément de l'information

séquentielle est bénéfique. Dans ce sens, nous proposerons d'incorporer à notre modélisation originale des variables liées à la densité structurelle locale, qui, à notre connaissance, n'ont pas encore été intégrées aux approches de modélisation dans la littérature.

Approches tridimensionnelles

Dans ce cadre méthodologique, la modélisation des sites de phosphorylation intègre également des acides aminés voisins définis selon leur proximité spatiale. Cette approche est rendue possible par l'utilisation des coordonnées tridimensionnelles des atomes au sein des acides aminés. Ces coordonnées peuvent être récupérées dans la base de données Alphafold [30]. Yu et al. (2024) [31] s'appuient sur les coordonnées de l'atome de carbone alpha, l'atome central et commun à tous les acides aminés, afin de déterminer leur localisation au sein des protéines. Ils établissent ensuite un seuil de distance fixe et tout acide aminé situé à une distance inférieure à celui-ci est considéré comme étant en contact avec le site de phosphorylation. De plus, Yu et al. (2024) exploitent ces coordonnées afin de calculer des angles de torsion et ainsi représenter l'orientation spatiale du site de phosphorylation. Les résultats obtenus avec cette approche sont nettement supérieurs aux performances des modèles revus précédemment. Néanmoins, Yu et al. (2024) utilisent des réseaux de neurones convolutifs profonds, qui, en raison de leur architecture complexe et de leur fonctionnement en tant que « boîtes noires », rendent difficile l'interprétation des résultats. Nous montrerons qu'en améliorant la robustesse de la modélisation tridimensionnelle, il devient possible d'utiliser des méthodes de prédiction plus simples, permettant de préserver l'interprétabilité, sans compromettre les performances du modèle.

Synthèse des approches de modélisation

En résumé, elles peuvent être classées en trois catégories : séquentielle, structuro-séquentielle et tridimensionnelle. Les approches séquentielles utilisent une fenêtre d'acides aminés adjacents au site de phosphorylation et se subdivisent en deux groupes : les méthodes classiques qui intègrent explicitement les caractéristiques physico-chimiques et les méthodes avec jetons qui utilisent des représentations latentes. Les méthodes structuro-séquentielles ajoutent des informations sur la structure locale, telles que le score de désordre, la surface accessible au solvant et la structure secondaire. Les approches tridimensionnelles enrichissent leur modélisation avec les coordonnées spatiales des atomes. Nous avons également indiqué que la littérature propose des modèles généraux ainsi que des modèles spécifiques aux kinases, ce qui rend plus difficile la comparaison équitable des performances.

Notre méthode de modélisation originale est tridimensionnelle et propose une méthodologie

plus robuste pour agréger les propriétés des acides aminés environnants. La figure 2.1 permet de visualiser l'ensemble des méthodes revues et la case colorée représente où se situe notre approche. Pour les approches structuro-séquentielles et tridimensionnelles, nous intégrons dans la figure la distinction entre les approches classiques et celles utilisant des jetons, puisqu'elles comportent aussi une composante séquentielle pouvant adopter l'une de ces deux formes.

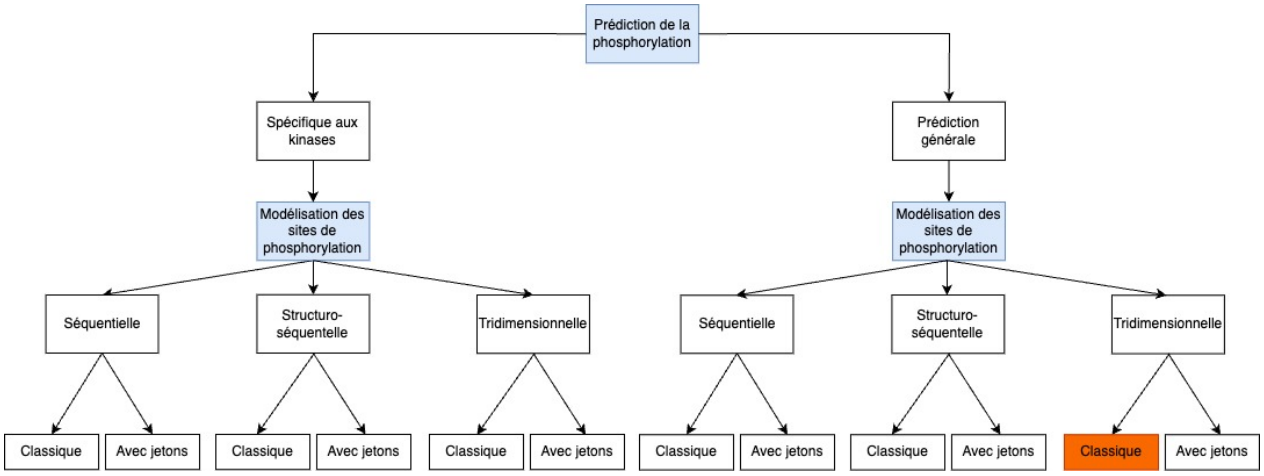

FIGURE 2.1 Diagramme des méthodes de modélisation des modèles de prédiction de la phosphorylation.

2.1.2 Méthodes d'échantillonnage des séquences d'acides aminés

Tel que souligné précédemment, dans les jeux de données sur la phosphorylation, le nombre de résidus non phosphorylés excède largement celui des sites phosphorylés. Ce déséquilibre peut conduire les modèles à privilégier systématiquement la classe majoritaire, réduisant ainsi leur capacité à détecter la classe minoritaire, les acides aminés phosphorylés dans notre cas. De plus, la grande quantité d'observations négatives entraîne une redondance importante dans les données. Pour remédier à ces problèmes de déséquilibre et de redondance, la grande majorité des modèles dans la littérature utilisent des méthodes telles que CD-HIT [32] et BLASTClust [33]. Ces méthodes établissent un seuil de similarité (généralement entre 40% et 70%) afin d'exclure les séquences trop semblables. La similarité est calculée en déterminant la proportion d'acides aminés identiques présents aux positions correspondantes dans les séquences comparées.

Hai Dang et al. (2008) [11] ainsi que Gao et al. (2010) [21] proposent des approches d'échantillonnage spécifiquement conçues pour la prédiction de la phosphorylation. Hai Dang et al. (2008) emploient en premier lieu une méthode alternative proposée par Kim, J.H. et al. (2004) [34] afin de regrouper les sites de phosphorylation en fonction des familles de kinases

impliquées. Après cette étape, les auteurs établissent des intervalles de confiance à partir de la distribution des observations de la classe positive. Ainsi, Hai Dang et al. (2008) peuvent sélectionner des échantillons d'acides aminés pour lesquels ils disposent d'une plus grande certitude quant à leur appartenance réelle à la classe négative, adressant directement cette incertitude au sein des données. Cependant, cette approche ne peut pas s'appliquer aux modèles généraux, car elle fait usage de la méthode de regroupement développée par Kim, J.H. et al. (2004), basée sur les familles de kinases. Gao et al. (2010) proposent une méthode originale basée sur le « bootstrap aggregating » pour gérer le déséquilibre des classes après avoir préalablement réduit les séquences à l'aide de BLASTClust avec un seuil de similarité de 50%. Bien que cette technique soit robuste, elle reste dépendante d'une réduction préalable des données basée sur la similarité séquentielle.

En conclusion, les méthodes CD-HIT et BLASTClust, malgré le fait qu'elles soient utilisées par la majorité des modèles revus, présentent des limitations car elles calculent la similarité entre séquences uniquement sur la base d'une représentation linéaire. En revanche, notre approche originale d'échantillonnage incorpore l'aspect tridimensionnel des séquences, permettant ainsi un échantillonnage plus diversifié des données, améliorant par conséquent la capacité de généralisation de notre modèle. De plus, contrairement à la plupart des méthodes de la littérature, notre approche tient explicitement compte de l'incertitude liée à l'appartenance de certains acides aminés à la classe négative.

2.1.3 Sélection de variables dans un contexte de prédiction de la phosphorylation

On peut séparer les méthodes de sélection de variables en deux grandes catégories : la réduction de dimension et la sélection. En réduction de dimension, on construit un espace latent qui résume l'information issue des relations entre les variables d'origine. En sélection, on réduit le nombre de variables en sélectionnant les plus pertinentes selon divers critères. L'approche de sélection semble être privilégiée dans les modèles de la littérature [14–17, 20]. Cela peut s'expliquer par l'importance de conserver l'interprétabilité des résultats, ce qui est plus difficile à faire si des variables latentes sont employées. Puisque la capacité d'interpréter les résultats est un élément important dans notre contexte, l'accent est mis sur les méthodes de sélection pour cette partie de la revue de littérature.

Basu et al. (2010) [15] filtrent les variables par analyse de corrélation. Cela permet de réduire le nombre de variables dans le modèle, contribuant à faciliter son interprétabilité. Cependant, il s'agit d'une méthode univariée, pouvant avoir de la difficulté à capturer les relations entre les variables. Ahmed et al. (2021) [20] utilisent également une méthode univariée, le score-F,

qui estime le pouvoir discriminatif des caractéristiques en calculant le score de Fisher à partir des données d'apprentissage.

Hamid et al. (2016) [16] font usage du coefficient de Gini pour identifier et retenir les variables avec le plus d'impact sur la performance. Cela leur a permis de classer les caractéristiques de la plus influente à la moins influente, et de fixer le nombre final de variables à inclure dans leur modèle. Toutefois, Strobl et al. (2007) [35] ont montré que le coefficient de Gini a tendance à favoriser les variables continues ayant une échelle plus grande et les variables discrètes avec une grande cardinalité. En effet, celles-ci offrent plus de points de division possibles dans les arbres de décision, augmentant ainsi artificiellement leur importance par rapport aux variables continues ayant une petite échelle et aux valeurs discrètes à faible cardinalité.

La sélection avant et la sélection arrière sont deux approches qui semblent revenir fréquemment dans la littérature. À titre d'exemple, Biswas et al. (2010) [14] procèdent à une sélection avant qui ne commence qu'avec un seul prédicteur et qui, de manière itérative, ajoute des prédicteurs un à un. À chaque itération, l'algorithme évalue toutes les variables candidates restantes et sélectionne celle dont l'inclusion réduit le plus l'erreur. Il est également possible d'utiliser d'autres critères comme l'information d'Akaike (AIC), le critère d'information bayésien (BIC) et le R^2 ajusté, selon la méthode de prédiction. Le processus itératif se poursuit jusqu'à ce que l'ajout de nouvelles variables n'apporte pas d'amélioration significative ou jusqu'à ce qu'un critère d'arrêt soit rempli. Les auteurs obtiennent donc un sous-ensemble de variables avec des performances qu'ils jugent satisfaisantes. Cependant, lors du processus de sélection avant, l'ajout d'une variable peut rendre une autre variable non significative, sans la retirer du modèle, ce qui peut potentiellement entraîner la sélection d'un nombre de variables plus grand que nécessaire. Nous réutiliserons d'ailleurs le concept de critère d'arrêt dans notre méthode originale de sélection de variables, mais sans effectuer une sélection avant.

Wei et al. (2017) [17] utilisent la sélection arrière, consistant à initialiser un modèle avec toutes les variables, puis à les retirer une à une. Grâce à cette méthode, ils identifient les caractéristiques dont la suppression provoque une baisse significative de la performance. Toutefois, lorsqu'une variable est éliminée, elle n'est plus réintroduite, ce qui empêche la détection d'interactions potentielles. Si l'on souhaite considérer les interactions potentielles, il faut explorer toutes les combinaisons possibles de variables, ce qui peut rapidement devenir très lourd sur le plan computationnel, en raison de l'explosion combinatoire de solutions possibles. Dans ce sens, nous verrons que notre méthode originale propose de limiter l'exploration de solutions à un sous-ensemble plus prometteur, en utilisant des probabilités a priori.

Nous notons également que les modèles dans la littérature n'intègrent pas tous forcément une

phase de sélection de variables. En effet, ceux exploitant le mécanisme d'attention [12, 13, 18, 19, 22, 23, 31] ne procèdent pas à cette étape. Ces modèles utilisent des variables latentes qui ne peuvent pas être interprétées d'un point de vue biologique, rendant ainsi l'étape de sélection de variables superflue, puisque l'objectif est d'augmenter l'interprétabilité.

GasPhos [36] est un modèle proposé par Chen et al. (2020), dans lequel les auteurs combinent 2 métaheuristiques, les colonies de fourmis et un algorithme génétique, pour sélectionner un sous-ensemble de caractéristiques. Les auteurs ont démontré que l'utilisation de métaheuristiques permettait d'atteindre de meilleures performances que des méthodes univariées comme l'analyse par corrélation et l'information mutuelle. Le modèle GasPhos est cependant un modèle spécifique aux kinases, leur méthode de sélection de variables est donc conçue pour être intégrée à ce genre de modèle. De plus, GasPhos a été créé pour prédire la phosphorylation provoquée par 6 kinases spécifiques, et on estime à plus de 500 le nombre de kinases pouvant phosphoryler la sérine, thréonine ou la tyrosine dans le corps humain [37].

2.1.4 Synthèse de la revue de littérature des modèles de prédiction de la phosphorylation

Pour clore cette section, l'approche séquentielle semble être la plus fréquemment adoptée pour modéliser les sites de phosphorylation, mais il a été montré qu'intégrer une perspective tridimensionnelle est bénéfique. En effet, les approches tridimensionnelles permettent de considérer l'influence d'acides aminés proches d'un point de vue spatial, mais distants sur le plan séquentiel. Nous démontrerons qu'il est possible d'améliorer les performances de prédiction en augmentant la robustesse de la modélisation tridimensionnelle. Concernant les méthodes d'échantillonnage, CD-HIT et BLASTClust sont employées par la majorité des modèles. Cependant, elles ne considèrent pas l'aspect tridimensionnel dans le calcul de similarité entre les séquences, ni l'incertitude liée à l'appartenance de certains acides aminés à la classe négative. Nous proposerons donc une méthode originale de sous-échantillonnage de la classe négative qui aborde directement ces deux limitations. Enfin, les approches de sélection de variables univariées sont couramment utilisées, mais elles ont une capacité limitée à capturer des relations potentielles entre les variables. Chen et al. (2020) ont démontré qu'il est possible d'adapter des métaheuristiques, permettant de capturer les relations potentielles entre les variables et d'explorer une variété de solutions possibles. Dans ce sens, notre approche originale de sélection de variables intègre des connaissances biologiques spécifiques à la phosphorylation et fait usage de probabilités a priori pour guider la recherche de solutions, menant à une réduction du nombre de variables avec un impact minimal sur les performances. Le tableau 2.1 présente les 17 modèles examinés dans cette revue de littérature, ainsi que les

différentes méthodes de modélisation, d'échantillonnage et de sélection de variables, en plus des méthodes de prédiction.

TABLEAU 2.1 Méthodes de modélisation, de prédiction, de sélection de variables et d'échantillonnage des modèles de prédiction de la phosphorylation d'acide aminés revus.

Type de prédiction	**Nom**	**Modélisation**	**Méthode de prédiction**	**Échantillonnage**	**Sélection de variables**	**Année**	**Réf.**
Spécifique aux kinases	CRPhos	Séquentielle	Champs de Markov	Familles de kinase	Non spécifié	2008	[11]
	GasPhos	Structuro-séquentielle	Forêt aléatoire	CD-HIT	Algo. gén. et col. de fourmis	2020	[36]
	GPS 6.0	Structuro-séquentielle	RNA, LightGBM	CD-HIT	Non spécifié	2023	[38]
	DL-SPhos	Séquentielle	Transformeur	CD-HIT	Non spécifié	2024	[13]
Générale	PPREDD	Séquentielle	Machines vecteurs supp.	Non spécifié	Sélection avant	2010	[14]
	AMS 3.0	Séquentielle	RNA	Non spécifié	Corrélation et sélection avant	2010	[15]
	PhosphoSVM	Structuro-séquentielle	Machines vecteurs supp.	BLASTClust	Non spécifié	2014	[26]
	RF-Phos 2.0	Séquentielle	Forêt aléatoire	skipredundant	Coefficient de Gini	2016	[16]
	iPhos-PseEn	Structuro-séquentielle	Forêt aléatoire	Non spécifié	Non spécifié	2016	[25]
	PhosPred-RF	Séquentielle	Forêt aléatoire	CD-HIT	Sélection arrière	2017	[17]
	Transphos	Séquentielle	Transformeur	BLASTClust	Non spécifié	2022	[18]
	Attenphos	Séquentielle	Attention, RNC	CD-HIT	Non spécifié	2024	[19]
	DeepPPSite	Séquentielle	LSTM	CD-HIT	Score-F	2021	[20]
Hybride	Musite	Séquentielle	Machines vecteurs supp.	BLASTClust, méthode originale	Non spécifié	2010	[21]
	MusiteDeep	Séquentielle	Attention, RNC	Non spécifié	Non spécifié	2017	[22]
	DeepPhos	Séquentielle	RNC, CNN	CD-HIT	Non spécifié	2019	[23]
	Phos-AF	Tridimensionnelle	Attention, CNN	CD-HIT	Non spécifié	2024	[31]

CHAPITRE 3 PRÉDICTION DES SITES DE PHOSPHORYLATION

3.1 Définition du problème

L'objectif de ce chapitre est de développer un modèle de prédiction de la phosphorylation d'acides aminés plus précis que ceux retrouvés dans la littérature, et possédant un meilleur niveau d'interprétabilité. Prédire la phosphorylation des acides aminés représente un défi majeur et persistant dans la communauté scientifique. En effet, le nombre conséquent de modèles dans la littérature [11, 13–23, 25, 26, 31, 36, 38] et l'intérêt soutenu de la communauté scientifique pour ce sujet depuis près de vingt ans en témoignent. Pour ce problème, les techniques d'apprentissage machine sont particulièrement bien adaptées, et ce, pour deux raisons principales. Premièrement, d'importants volumes de données sont disponibles pour modéliser le phénomène de la phosphorylation et entraîner les algorithmes. Deuxièmement, ce problème s'inscrit dans un contexte bio-moléculaire, caractérisé par des relations complexes entre les variables, un cadre où l'apprentissage machine brille particulièrement.

Tel que souligné, il existe une grande quantité de données à notre disposition. Nous les avons classées en 2 grandes catégories : la première décrit directement les événements de phosphorylation sur les acides aminés, tandis que la deuxième rassemble de l'information sur les propriétés des acides aminés. En particulier, la première catégorie de données compile des résultats d'études expérimentales visant à valider la phosphorylation de résidus de sérine, de thréonine et de tyrosine spécifiques. Les données se présentent sous la forme de listes d'acides aminés, au sein de différentes protéines, en y indiquant si leur phosphorylation a été validée expérimentalement ou non. Dans le cadre de notre projet, nous utilisons l'ensemble de données Phospho.ELM (PELM) [39], puisque celui-ci est spécifique aux protéines humaines. La deuxième catégorie regroupe les ensembles que nous qualifions de « connexes ». Ceux-ci ne traitent pas directement des sites de phosphorylation, mais sont utiles pour la modélisation de ces derniers. En effet, les bases de données sur la phosphorylation, comme PELM, ne contiennent pas l'information contextuelle (structurelle ou physico-chimique) nécessaire pour modéliser les sites de phosphorylation. La très grande variété de données connexes pouvant être employées dans notre modélisation et le fait que les mécanismes biologiques sous-jacents ne sont pas tous connus à ce jour rendent notre problème particulièrement complexe. En effet, il ne s'agit pas d'un contexte où les caractéristiques de chaque observation nous sont fournies à l'avance. Par conséquent, un travail important en amont est nécessaire pour comprendre les mécanismes de la phosphorylation, ce qui permettra ensuite de modéliser adéquatement nos observations. Ainsi, les bases de données connexes que nous avons employées sont :

PubChem [40], pour les propriétés physico-chimiques des acides aminés, et AlphaFold [30], pour intégrer leur aspect tridimensionnel.

Bien que la présentation détaillée de ces ensembles de données soit réservée à une section ultérieure, il est pertinent de souligner, dès maintenant, les problèmes dont souffre l'ensemble de données PELM. Premièrement, PELM présente un déséquilibre important : les résidus pour lesquels la phosphorylation a été validée ne représentent que 4% (26 650) du nombre total d'observations (680 659). De plus, un résidu étiqueté comme « non phosphorylé » n'est pas nécessairement incapable de l'être ; son statut peut simplement être dû à l'absence d'une validation expérimentale.

Un autre défi lié à notre problématique découle du fait que notre modèle s'inscrit dans une démarche d'identification de biomarqueurs, où la capacité d'interpréter les résultats de prédiction est importante. Afin que les résultats soient interprétables biologiquement, il faut éviter l'utilisation de méthodes de style « boîte noire », dans lesquelles il est difficile de retracer le processus menant aux prédictions. Il faut également limiter les variables incluses aux plus pertinentes, afin de faciliter l'analyse biologique de leur impact sur les prédictions, et ce, sans nuire aux performances prédictives.

Face à ces défis, plusieurs techniques d'intelligence artificielle s'offrent à nous. Parmi celles-ci figurent l'apprentissage supervisé, où le modèle apprend à partir de données étiquetées, l'apprentissage non supervisé, qui vise à découvrir des patrons dans des données non étiquetées, et l'apprentissage semi-supervisé, qui combine les deux. Dans le cadre de ce projet, étant donné que l'état de phosphorylation est connu pour un certain nombre de résidus, nous adoptons principalement une approche supervisée pour faire nos prédictions. Cependant, pour adresser le déséquilibre des classes et l'incertitude liée à la classe négative, nous intégrerons une approche non supervisée à notre méthode d'échantillonnage.

Un modèle d'apprentissage machine peut être conçu pour effectuer une tâche de classification ou de régression. En classification, l'objectif est de prédire à quelle classe une observation appartient, en utilisant une variable discrète pour y faire référence. En régression, on cherche plutôt à prédire la valeur d'une variable continue. Dans notre projet, afin de prédire l'état de phosphorylation d'un résidu, nous utiliserons la classification à 2 classes.

Afin d'atteindre notre objectif, nous avons structuré la suite de ce chapitre comme suit. Nous présentons tout d'abord une méthode originale de modélisation tridimensionnelle des sites de phosphorylation. S'ensuit la présentation de l'approche systématique d'apprentissage machine, qui inclut notamment une méthode originale d'échantillonnage et une méthode originale de sélection de variables.

3.2 Modélisation des sites de phosphorylation

Nous traitons le problème de prédiction de la phosphorylation d'acides aminés comme un problème de classification binaire. Ainsi, un résidu de sérine, de thréonine ou de tyrosine peut appartenir à l'une des deux classes suivantes : « phosphorylé » ou « non phosphorylé ». Afin de représenter un résidu et son état de phosphorylation, nous employons la notation suivante : $(\mathbf{x}_{a,s}; y_{a,s})$, où $y_{a,s}$ est définie comme suit :

$$y_{a,s} = \begin{cases} 0, & \text{si le résidu } a \text{ n'est pas phosphorylé au sein de la protéine } s \\ 1, & \text{sinon.} \end{cases} \tag{3.1}$$

$\mathbf{x}_{a,s}$ est un vecteur contenant 223 variables, caractérisant l'acide aminé a au sein de s. Les 223 variables qui constituent $\mathbf{x}_{a,s}$ sont issues de notre méthode de modélisation. Pour faire référence à un résidu a, nous utilisons sa position au sein de sa protéine s, et nous employons l'identifiant standard de la protéine s pour faire référence à cette dernière. Pour donner un exemple, l'observation $(\mathbf{x}_{181,P10636}; y_{181,P10636})$ fait référence au résidu à la position 181 dans la protéine P10636. Afin de prédire l'état de phosphorylation d'un résidu a au sein de s, notre modèle estime la probabilité suivante :

$$P[y_{a,s} = 1 \mid \mathbf{x}_{a,s}]. \tag{3.2}$$

La présentation des éléments des vecteurs $\mathbf{x}_{a,s}$ est structurée en 3 parties, chacune étant reliée à un sous-groupe de caractéristiques. D'abord, nous détaillons les bases de données utilisées ainsi que les éléments retenus pour chacune d'elles. Lors de cette première étape, nous définissons les 13 premières variables s'insérant dans les vecteurs $\mathbf{x}_{a,s}$. Celles-ci décrivent le type des acides aminés voisins au résidu a. Ensuite, nous détaillons notre stratégie d'agrégation des données tridimensionnelles et des propriétés physico-chimiques, formant 192 variables supplémentaires. Finalement, nous présentons les 18 dernières variables liées à la densité de l'environnement local des sites de phosphorylation, complétant ainsi les 223 caractéristiques qui forment nos vecteurs d'entrée $\mathbf{x}_{a,s}$.

3.2.1 Bases de données et intrants sélectionnés

La première base de données, PELM [39], porte sur les données de phosphorylation. La seconde, PubChem [40], fournit les propriétés physico-chimiques pour chacun des 20 types d'acides aminés différents. La troisième, Alphafold [30], contient les coordonnées spatiales des atomes des acides aminés qui composent les protéines de PELM.

Données de phosphorylation

Pour notre étude, nous avons considéré 3 bases de données portant sur la phosphorylation : PhosphoSitePlus, Database of Post-Translational Modifications et PELM. Tel que souligné précédemment, notre choix s'est porté sur cette dernière, car elle est spécifiquement dédiée aux protéines humaines. Nous en avons extrait deux éléments, détaillés ci-bas.

La première information que nous avons extraite est l'état de phosphorylation, $y_{a,s}$, de chaque résidu de sérine, thréonine et tyrosine à travers les 8 841 différentes protéines présentes. Le tableau 3.1 fournit des statistiques globales de phosphorylation pour les résidus d'intérêt dans l'ensemble de données.

Tableau 3.1 Statistiques de phosphorylation de Phospho.ELM.

Type / **Classe**	Sérine	Thréonine	Tyrosine	**Total**
Phosphorylé	19 429	4 827	2 394	**26 650**
Non phosphorylé	342 597	213 234	98 178	**654 009**
Total	**362 026**	**218 061**	**100 572**	**680 659**

Rappelons que même si la phosphorylation d'un résidu appartenant à l'un de ces 3 types n'a pas été validée expérimentalement, celui-ci peut quand même être sujet à cette réaction chimique. Remarquons également le débalancement important entre les classes. Ces 2 problèmes seront adressés dans le cadre de notre méthode originale d'échantillonnage.

La seconde information retenue de PELM est une fenêtre séquentielle de 13 résidus, indiquant le type de l'acide aminé d'intérêt ainsi que celui de ses 6 voisins en amont et en aval. Nous justifions le choix relatif à la taille de cette fenêtre avec 3 arguments. Le premier est de nature biologique, le second est lié à l'interprétabilité et le troisième concerne la performance prédictive. Le premier argument biologique s'appuie sur les travaux de Bradley et al. (2019) et Brinkworth et al. (2002), indiquant que les préférences des kinases pour les motifs d'acides aminés voisins à leur cible sont limitées à une fenêtre de 9 résidus ($\pm$ 4). À titre d'exemple, pour les kinases d'une certaine famille, les kinases dépendantes des cyclines (KDC), une condition nécessaire, mais non suffisante, afin de catalyser la phosphorylation d'une sérine est que celle-ci soit immédiatement suivie d'un type d'acide aminé spécifique, la proline [41]. Concrètement, cela signifie que si le résidu de sérine est situé à une position donnée dans sa protéine s, 120 par exemple, le résidu de proline doit être situé à la position 121. Chaque protéine, également appelée chaîne protéique, est en effet composée d'un nombre L

d'acides aminés ordonnés selon une séquence standard établie par la communauté scientifique, allant du premier (position 1) au dernier (position L). Pour illustrer l'influence des KDC sur l'ensemble de données PELM, la figure 3.1 montre la distribution des acides aminés qui suivent directement les résidus de sérine phosphorylés, appelés phosphosérines. Le deuxième

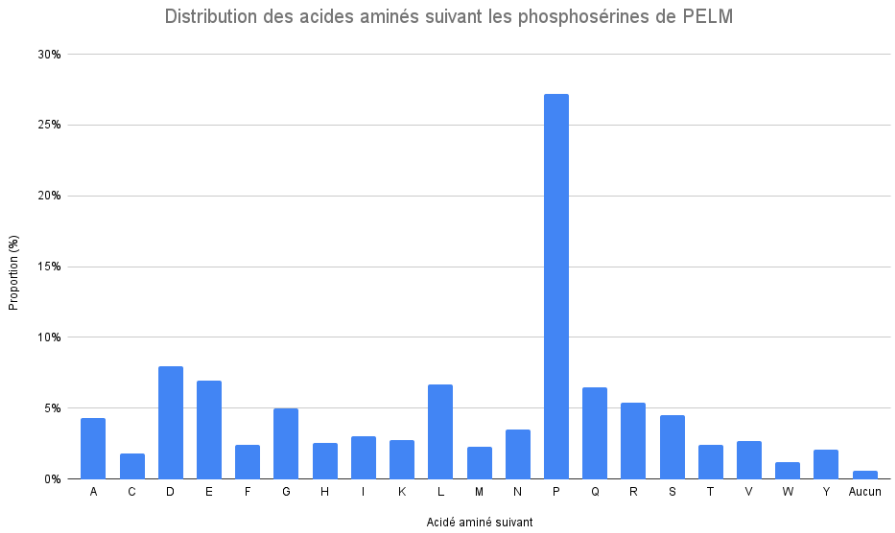

FIGURE 3.1 Distribution des types d'acides aminées suivant les phosphosérines de PELM. « Aucun » représente le cas où la phosphosérine se retrouve à la fin de sa chaîne protéique.

argument, lié à l'interprétabilité, tient au fait qu'une fenêtre trop large compromettrait la lisibilité des résultats. En effet, la littérature semble indiquer que les motifs reconnus par les kinases ne dépassent pas 4 résidus en amont ou en aval du résidu cible. Par conséquent, inclure des acides aminés éloignés de 10, 15 ou même 20 positions introduirait un bruit inutile, nuisant à la capacité d'interprétation des scientifiques susceptibles d'utiliser notre modèle.

Notre troisième et dernier argument s'appuie sur l'analyse du gain marginal de performance, obtenu en élargissant progressivement la fenêtre séquentielle fournie en entrée à un modèle de prédiction de la phosphorylation des résidus de PELM. L'objectif de cet exercice est de voir dans quelle mesure l'accroissement de la taille de la fenêtre influence les performances prédictives. Pour chaque observation, le modèle n'utilise en entrée qu'un vecteur contenant l'identité des acides aminés voisins. Nous avons utilisé une régression logistique et une analyse par discriminant linéaire (LDA) pour faire les prédictions. Nous avons choisi ces modèles car ils comportent peu d'hyperparamètres, en faisant d'eux de bons candidats pour cette étape intermédiaire. En étudiant les performances pour des tailles de fenêtres allant de 3 à 65, nous avons constaté que le gain apporté par l'ajout d'un acide aminé fluctuait autour de zéro au-delà de 13 résidus, et ce, pour les 2 modèles. Le graphique des gains marginaux pour

le modèle de régression logistique est présenté à la figure 3.2 et celui du modèle LDA est disponible dans l'annexe A.

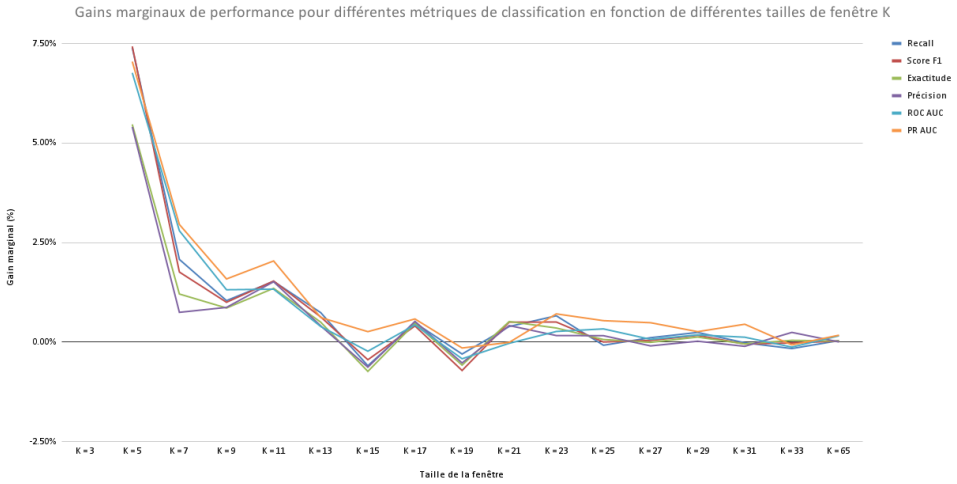

FIGURE 3.2 Gains marginaux de performance du modèle de régression logistiques en fonction de différentes tailles de fenêtre K.

Compte tenu de la combinaison des 3 arguments présentés ci-haut, nous jugeons approprié d'employer 13 comme taille finale de fenêtre. Ainsi, l'information liée à la fenêtre séquentielle d'un résidu prend la forme d'un ensemble, noté $SEQ_{a,s}$, composé de 13 variables discrètes et défini comme suit : $SEQ_{a,s} = \{Type_{a-6}, ..., Type_{a-1}, Type_a, Type_{a+1}, ..., Type_{a+6}\}$, où chaque variable représente le type de l'acide aminé à la position correspondante. Chaque élément de $SEQ_{a,s}$ peut prendre une valeur parmi celles de l'ensemble $Types$, qui contient les codes des 20 types d'acides aminés, et est défini comme suit : $Types$ = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y , $\emptyset$ }, où $\emptyset$ représente le cas où l'acide aminé à la position correspondante est inexistant. En effet, cette situation se produira si le résidu central a est parmi les 5 premiers ou 5 derniers acides aminés de sa chaîne protéique s.

Ainsi, pour un résidu a au sein d'une protéine s, nous extrayons de PELM son état de phosphorylation, $y_{a,s}$ et sa fenêtre séquentielle $SEQ_{a,s}$. Les 13 variables discrètes des fenêtres $SEQ_{a,s}$ sont donc intégrées aux vecteurs $\mathbf{x}_{a,s}$. La figure 3.3 offre une visualisation de la fenêtre $SEQ_{a,s}$ d'un résidu a au sein d'une protéine s constituée de L acides aminés.

Propriétés physico-chimiques des acides aminés

La structure moléculaire des acides aminés est constituée de 3 composantes principales : un groupe amine, un groupe acide carboxylique ainsi qu'une chaîne latérale. Chacun des 20 types

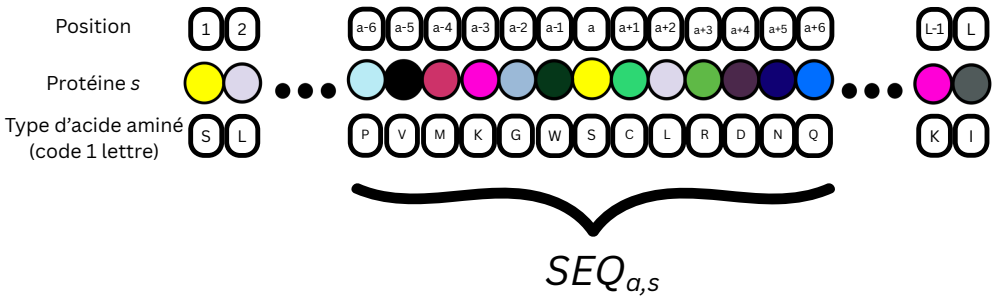

FIGURE 3.3 Illustration de la fenêtre séquentielle $SEQ_{a,s}$.

d'acides aminés possède une chaîne latérale unique, alors que les deux autres composantes sont identiques d'un type à l'autre. La figure 3.4 montre la structure globale des acides aminés, en y détaillant celles des résidus qui nous intéressent dans cette étude.

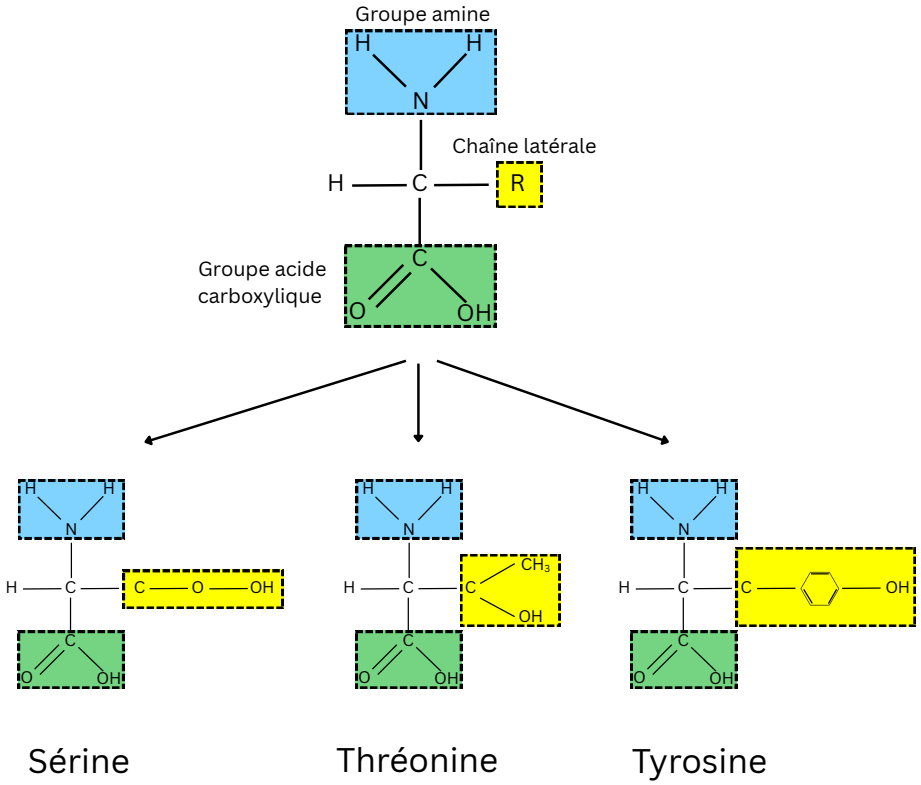

FIGURE 3.4 Figure mettant en relation la structure générale d'un acide aminé (haut) et les structures des chaînes latérales de la sérine, thréonine et tyrosine (bas).

On retrouve dans la figure 3.4 les groupes hydroxyles (-OH) dans les chaînes latérales, en jaune. C'est à cet endroit que le groupement phosphate (PO_3^{2-}) s'ajoute lors de la phosphorylation. Afin que cette réaction chimique soit possible, le groupe hydroxyle doit être

accessible et prêt à réagir avec la kinase qui catalyse le processus. On observe dans la figure 3.4 que dans le cas de la sérine, ce groupe hydroxyle est bien dégagé et lié seulement à un atome de carbone. À l'inverse, dans le cas de la tyrosine, le groupe hydroxyle est attaché à une structure en anneau, qui attire fortement vers elle les électrons, particules élémentaires chargées négativement, appartenant au groupe hydroxyle. Ce « vol » d'électrons a pour effet de limiter la réactivité chimique du groupe hydroxyle de la tyrosine. Par conséquent, pour que la phosphorylation se produise, ce groupe nécessite une interaction très spécifique, et donc plus rare, avec une kinase. Ce genre de subtilité entre les chaînes latérales est ce qui influence la dynamique de phosphorylation des résidus.

On peut caractériser les chaînes latérales en fonction de 3 attributs : l'uniformité de la distribution de leurs charges électriques (polarité), leur tendance à interagir avec l'eau (profil hydrophobe/hydrophile) ainsi que leur nature chimique (acide/basique). À partir de ceux-ci, les acides aminés sont regroupés en cinq catégories distinctes :

1. Non polaires et hydrophobes : Les charges électriques de ces chaînes latérales sont uniformément réparties, les rendant moins réactives. Elles auront également tendance à fuir l'eau. Les acides aminés avec de telles chaînes sont la glycine, l'alanine, la valine, la leucine, l'isoleucine, la méthionine, la phénylalanine, le tryptophane et la proline.
2. Polaires hydrophobes : La configuration de la chaîne latérale de ces résidus les empêche de rentrer en contact avec l'eau. Malgré leur polarité, leur caractère hydrophobe font que ces chaînes sont moins réactives. La cystéine et la tyrosine appartiennent à ce groupe.
3. Polaires hydrophiles : Les chaînes latérales de ces acides aminés ont tendance à interagir avec l'eau et sont très réactives. La sérine, la thréonine, l'asparagine et la glutamine se trouvent dans cette catégorie.
4. Acides : Les chaînes latérales de l'aspartate et du glutamate sont caractérisées par leur forte acidité. Cela leur confère des propriétés uniques leur permettant d'intervenir dans des réaction chimiques spécifiques.
5. Basiques : À l'inverse, les résidus avec des chaînes latérales basique incluent la lysine, l'arginine et l'histidine.

À la lumière de ce qui précède, on comprend que la polarité, le profil hydrophobe/hydrophile et la nature chimique des acides aminés constituent des éléments pertinents pour notre modélisation. Cependant, ces informations qualitatives manquent de granularité pour faire la distinction entre chacun des 20 types d'acides aminés. C'est précisément pourquoi nous employons les propriétés physico-chimiques : elles permettent de quantifier les différences subtiles qui existent entre les acides aminés, dues à l'unicité de leur chaîne latérale.

1. Coefficient de partage octanol-eau

Noté logP [42], le coefficient de partage octanol-eau est un indice variant de -4,2 à -1,1 selon le type d'acide aminé. Il décrit la tendance d'un résidu à se répartir entre un milieu hydrophobe (l'octanol) et un milieu hydrophile (l'eau). Ainsi, un environnement local constitué principalement de résidus à valeurs élevées de logP présentera un comportement hydrophobe. Les résidus situés dans un tel environnement chercheront donc à « fuir » l'eau, favorisant ainsi leur enfouissement au cœur de la protéine. À l'inverse, un environnement local hydrophile (faible logP) peut favoriser l'exposition du site de phosphorylation en surface, facilitant l'accès aux kinases.

2. Nombre de donneurs de pont hydrogène

Un pont hydrogène (pont H) est un type de liaison chimique entre deux molécules qui se fait en partageant un atome d'hydrogène. Dans ce genre de liaison, une molécule joue le rôle de donneur et l'autre de receveur. Le nombre de donneurs de pont H d'un acide aminé est étroitement lié à sa polarité. En effet, plus il en possède, plus il a la capacité de former des liaisons hydrogène avec d'autres molécules, ce qui entraîne une répartition inégale de ses charges électriques, caractéristique fondamentale de la polarité.

3. Nombre de receveurs de pont hydrogène

Un receveur de pont H est la molécule qui « reçoit » l'atome d'hydrogène. He et al. (2016) [43] ont montré qu'un environnement local composé de résidus possédant de nombreux receveurs de pont H favorise l'adoption par le segment protéique de conformations spécifiques qui facilitent l'accès aux kinases. En d'autres termes, puisque beaucoup d'acides aminés dans un tel environnement chercheront à se faire « donner » un atome d'hydrogène, afin de se stabiliser chimiquement, ils auront tendance à « s'exposer » le plus possible. Ainsi, compter le nombre de receveurs de pont H total dans un environnement local nous informe indirectement sur son accessibilité.

4. Nombre d'atomes lourds

La qualification d'un atome « lourd » dépend de son nombre de protons présents dans son noyau. Les protons sont des particules élémentaires qui composent les atomes et portent une charge positive. Le seuil exact de protons pour déterminer si un atome est lourd varie en fonction des molécules étudiées et de la variété des atomes qu'elles contiennent. Dans notre situation, la seule variation entre chaque acide aminé provient de la composition de leur chaîne

latérale. Celles-ci sont principalement constituées d'atomes de carbone, d'oxygène, d'azote et d'hydrogène. Dans ce contexte, la base de données PubChem définit un atome lourd comme tout atome différent de l'hydrogène. Les acides aminés avec un grand nombre d'atomes lourds possèdent des chaînes latérales plus encombrantes et sont donc plus susceptibles d'être enfouis dans le cœur de la protéine [44].

5. Surface polaire topologique

La surface polaire topologique (SPT) [45] d'un acide aminé est la somme de l'aire occupée par ses atomes polaires. Donc plus la SPT est grande, plus un acide aminé est polaire. La SPT peut également nous informer indirectement sur le profil hydrophobe/hydrophile d'un résidu. En effet, puisque les atomes polaires ont tendance à interagir avec l'eau, plus la SPT d'un acide aminé est grande, plus il aura une tendance hydrophile par effet de cascade. La SPT est mesurée en Ångström carré, noté $Å^2$, un Ångström équivaut à 1×10^{-10} mètre. Cette unité de mesure est fréquemment employée en physique atomique et nous en ferons usage lors de la représentation des acides aminés dans l'espace tridimensionnel.

6. Masse moléculaire

La masse moléculaire nous informe à quel point la chaîne latérale d'un acide aminé est imposante. Un résidu avec une petite chaîne latérale s'insère plus facilement à la surface des protéines, tandis qu'un acide aminé avec une chaîne plus lourde risque plutôt de s'enfouir au centre de la protéine [44]. La masse moléculaire est mesurée en Daltons (Da) et 1 Da équivaut à $1.66 \times 10^{-}27$ kilogramme.

7. Nombre de liaisons rotatives

Une liaison rotative est caractérisée par la liaison entre deux atomes d'hydrogène qui ne font pas partie d'une structure en forme d'anneau, comme on retrouve dans la chaîne latérale de la tyrosine (voir figure 3.4). Plus un acide aminé a de liaisons rotatives, plus sa chaîne latérale peut adopter des positions différentes [46]. Il faut se rappeler que la phosphorylation se produit sur le groupe hydroxyle de la chaîne latérale, donc si celle-ci peut se déplacer et pivoter sur elle-même plus librement, cela augmente sa capacité à bien se positionner pour une interaction avec une kinase. Un environnement caractérisé par des acides aminés ayant un grand nombre de liaisons rotatives peut donc contrebalancer le fait que ce même environnement soit enfoui au centre de la protéine, ce qui en fait une propriété intéressante.

8. Complexité moléculaire

Nous incluons cette propriété, car il existe un lien entre la complexité moléculaire des acides aminés et le degré de désordre des protéines [47]. En effet, les régions composées principalement d'acides aminés avec une haute complexité moléculaire sont plus susceptibles de se trouver dans les segments « désordonnés » de la protéine. Or, il a été démontré que les résidus situés dans ces régions désordonnées sont plus fréquemment phosphorylés que ceux présents dans des régions « stables » [48]. Proposée par Hendrickson et al. (1987) [49], la complexité d'un acide aminé est la somme de sa complexité structurale et chimique. Notée C, elle est définie comme suit :

$$C = C_\eta + C_E, \tag{3.3}$$

où C_η est la complexité structurale et C_E est la complexité liée à la diversité chimique. La complexité liée à la structure d'un acide aminé découle des types de liaisons qui lient ses atomes. La diversité chimique d'un acide aminé est, quant à elle, calculée à partir de la variété des types d'atomes qui le constituent.

Données spatiales des atomes au sein des acides aminés

Nous avons extrait les coordonnées (x,y,z) pour l'intégralité des atomes au sein des acides aminés qui composent les 8 841 protéines présentes dans l'ensemble de données PELM. En effet, nous ne sommes pas limités aux résidus de sérine, de thréonine et de tyrosine, puisque l'objectif est de modéliser l'environnement local de chacun de ces résidus dans sa globalité. Cependant, puisque les 8 propriétés physico-chimiques précédemment abordées concernent les acides aminés et non directement les atomes, nous avons dû mettre en place une stratégie permettant de passer des coordonnées atomiques aux coordonnées des acides aminés. Pour ce faire, nous avons utilisé la masse atomique et les coordonnées de chaque atome afin de déterminer le centre de masse de chaque résidu. Ce centre de masse définit ainsi la position de chaque résidu a dans chaque protéine s. Nous notons ces coordonnées comme suit : $c_{a,s} = (c^x_{a,s}; c^y_{a,s}; c^z_{a,s})$.

Notons $m_{i,a,s}$, la masse atomique de l'atome i au sein d'un acide aminé a constitué de n atomes, au sein d'une protéine s. Les coordonnées $(c^x_{a,s}; c^y_{a,s}; c^z_{a,s})$ du centre de masse de ce résidu au sein de s sont calculées selon les équations suivantes :

$$c^x_{a,s} = \frac{1}{m_{a,s}} \sum_{i=1}^{n} m_{i,a,s} x_{i,a,s} \tag{3.4}$$

$$c^{y}_{a,s} = \frac{1}{m_{a,s}} \sum_{i=1}^{n} m_{i,a,s} y_{i,a,s} \tag{3.5}$$

$$c^{z}_{a,s} = \frac{1}{m_{a,s}} \sum_{i=1}^{n} m_{i,a,s} z_{i,a,s}. \tag{3.6}$$

Dans les équations ci-haut, $m_{a,s} = \sum_{i=1}^{n} m_{i,a,s}$ et $(x_{i,a,s}; y_{i,a,s}; z_{i,a,s})$ désignent les coordonnées spatiales de l'atome i au sein de l'acide aminé a, appartenant à la chaîne protéique s.

Notre méthode représente plus fidèlement la position réelle des acides aminés dans les protéines que celle de Phos-AF [31], qui est d'ailleurs la seule autre approche utilisant des coordonnées spatiales identifiée dans la littérature. Cette autre approche se limite aux coordonnées de l'atome de carbone alpha pour définir la position des résidus dans l'espace. La figure 3.5 montre où se trouve cet atome au sein de la structure moléculaire d'un acide aminé spécifique, la tyrosine. Rappelons que chaque type de résidu possède une chaîne latérale unique, et donc composée d'atomes différents, et que celle-ci est critique pour les interactions intermoléculaires. Situer ces chaînes latérales dans l'espace avec plus de précision permet donc d'intégrer l'influence des résidus à proximité de manière plus robuste.

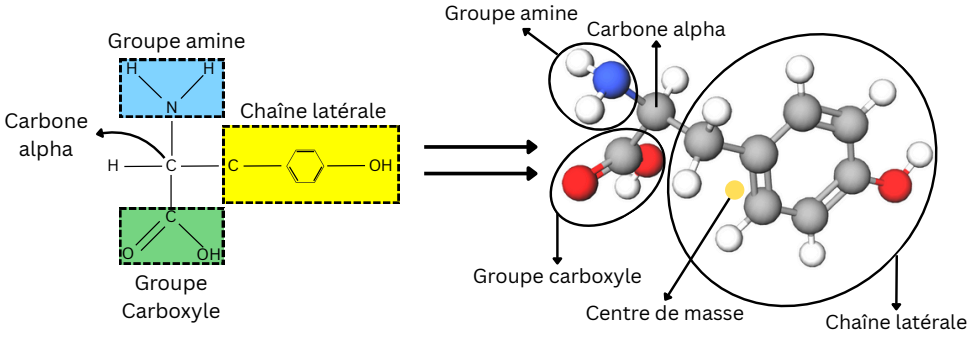

FIGURE 3.5 Structure moléculaire 2D et 3D de la tyrosine. Les atomes blancs représentent les atomes d'hydrogène, le rouge correspond aux atomes d'oxygène, le gris représente les atomes de carbone et le bleu est l'azote.

3.2.2 Agrégation des données spatiales et des propriétés physico-chimiques

Notre stratégie pour agréger les coordonnées spatiales et propriétés physico-chimiques des acides aminés mène à la création de 192 nouvelles caractéristiques. Celles-ci s'ajoutent aux

13 variables découlant de la fenêtre séquentielle $SEQ_{a,s}$, portant ainsi à 205 le nombre de variables définies jusqu'à présent, sur un total de 223. Nous présentons d'abord le choix de la taille de l'environnement local et sa division en 6 régions sphériques. Nous enchaînons ensuite avec la caractérisation de ces sphères via les propriétés physico-chimiques des acides aminés qu'elles contiennent.

Nous définissons en premier lieu 6 sphères de rayons distincts (R_1 = 5 Å, R_2 = 10 Å, R_3 = 20 Å, R_4 = 30 Å, R_5 = 40 Å, R_6 = 50 Å), centrées sur chacun des 680 659 résidus de sérine, thréonine et tyrosine. Cela représente donc un total de 4 083 954 régions sphériques (6 sphères/résidu × 680 659 résidus). La figure 3.5 offre une visualisation de ces 6 sphères pour un acide aminé au sein d'une protéine. Le choix concernant la taille des rayons repose sur

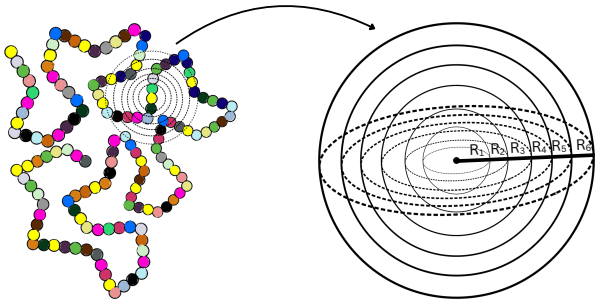

FIGURE 3.6 Illustration des 6 sphères de rayons R_1 à R_6 pour un résidu d'intérêt au sein d'une protéine. La figure n'est pas à l'échelle.

deux constats. D'une part, si l'on considère, pour chaque protéine humaine, une sphère de rayon minimal englobant tous ses acides aminés, le rayon médian de ces sphères est d'environ 20 Å [50]. D'autre part, la distance maximale observée entre deux acides aminés au sein d'une même protéine dans notre jeu de données est de 96 Å. Cela signifie donc qu'une sphère avec un diamètre de 96 Å, correspondant à un rayon de 48 Å, serait suffisante pour englober chaque protéine de notre ensemble de données.

Nous caractérisons ensuite l'environnement local de chaque résidu cible a (sérine, thréonine ou tyrosine) dans sa protéine s, en utilisant les 6 sphères définies précédemment et chacune des 8 propriétés physico-chimiques retenues. Pour ce faire, nous appliquons 4 méthodes d'agrégation sur les propriétés physico-chimiques des acides aminés contenus dans chaque sphère, ce qui explique les 192 variables (6 sphères × 8 propriétés physico-chimiques × 4 méthodes d'agrégation). Pour un résidu cible a dans une protéine s, et pour une sphère de rayon R_k ($k \in \{1, ..., 6\}$), nous définissons l'ensemble des résidus voisins $N_{a,s}(R_k)$ comme

suit :

$$N_{a,s}(R_k) = j \in \text{protéine } s \mid j \neq a \text{ et } d(c_{a,s}, c_{j,s}) < R_k, \tag{3.7}$$

où $c_{a,s}$ et $c_{j,s}$ sont les coordonnées du centre de masse des résidus a et j respectivement, tandis que $d(c_{a,s}, c_{j,s})$ est la distance euclidienne entre eux :

$$d(c_{a,s}, c_{j,s}) = \sqrt{(c^x_{a,s} - c^x_{j,s})^2 + (c^y_{a,s} - c^y_{j,s})^2 + (c^z_{a,s} - c^z_{j,s})^2}. \tag{3.8}$$

Les 8 propriétés physico-chimiques d'un résidu j sont notées $p_{j,l}$, où $l \in \{1, ..., 8\}$ indique une propriété spécifique. Les 4 méthodes d'agrégation, notées A_1, A_2, A_3, A_4, pour synthétiser l'information de chaque propriété physico-chimique l au sein de chaque sphère de rayon R_k autour d'un résidu a sont les suivantes :

1. A_1 : Somme simple de la valeur de la propriété l des voisins j dans la sphère :

$$\text{Somme}_l(a, s, R_k) = \sum_{j \in N_{a,s}(R_k)} p_{j,l} \tag{3.9}$$

2. A_2 : Somme pondérée par la distance inverse de la propriété l, donnant plus de poids aux résidus plus prés de a :

$$\text{SommePond}_l(a, s, R_k) = \sum_{j \in N_{a,s}(R_k)} w(d(c_{a,s}, c_{j,s}))p_{j,l} \tag{3.10}$$

3. A_3 : Moyenne simple de la propriété l pour les voisins j dans la sphère :

$$\text{Moyenne}_l(a, s, R_k) = \frac{1}{|Na, s(R_k)|} \sum_{j \in N_{a,s}(R_k)} p_{j,l} \tag{3.11}$$

4. A_4 : Moyenne pondérée par la distance inverse de la propriété l des voisins j dans la sphère :

$$\text{MoyennePond}_l(a, s, R_k) = \frac{\sum_{j \in N_{a,s}(R_k)} w(d(c_{a,s}, c_{j,s}))p_{j,l}}{\sum_{j \in N_{a,s}(R_k)} w(d(c_{a,s}, c_{j,s}))} \tag{3.12}$$

Dans les équations 3.10 et 3.12, $w(d(c_{a,s}, c_{j,s})) = \frac{1}{d(c,j)}$ est une fonction de pondération décroissante en fonction de la distance. L'application de ces 4 méthodes d'agrégation aux 8 propriétés physico-chimiques pour chacune des 6 sphères génère donc un total de $4 \times 8 \times 6 = 192$ caractéristiques pour chaque résidu cible a, que nous insérons dans leur vecteur d'entrée $\mathbf{x}_{a,s}$ respectif. La figure 3.6 offre une visualisation de notre stratégie d'agrégation des propriétés

physico-chimiques pour un résidu donné.

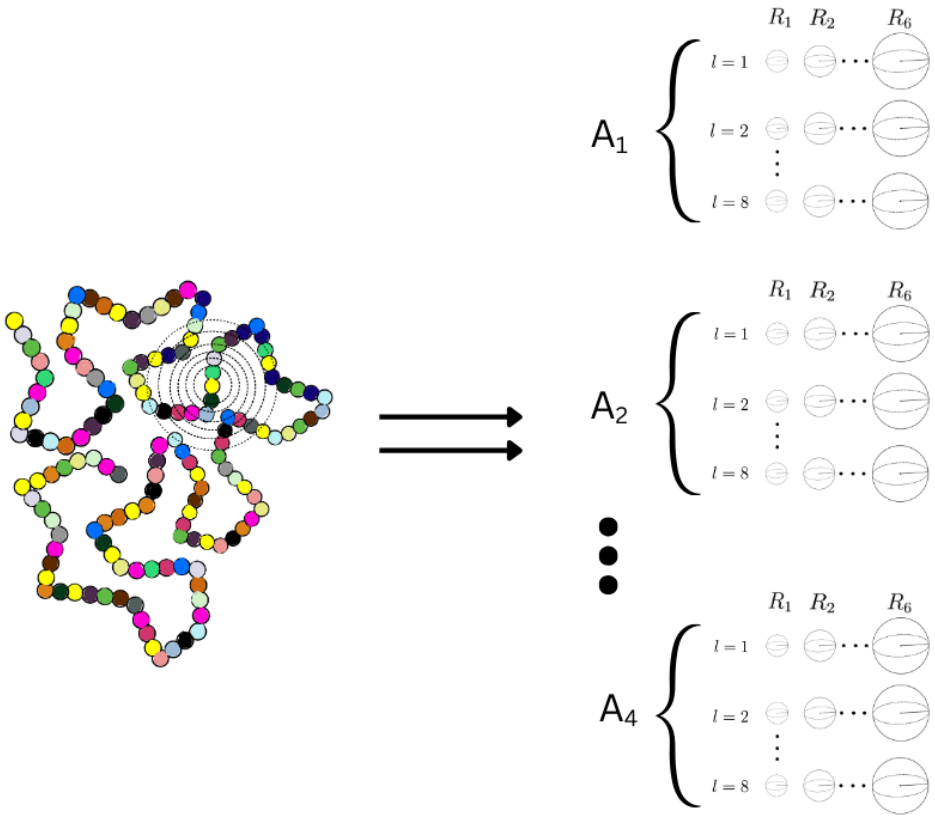

FIGURE 3.7 Illustration des 4 méthodes d'agrégation (A_1, A_2, A_3, A_4) appliquées aux 8 propriétés physico-chimiques l des acides aminés présents dans les 6 sphères de rayons R_1 à R_6 pour un résidu d'intérêt au sein d'une protéine.

3.2.3 Caractérisation de la densité locale

Nous complétons les vecteurs d'entrée $\mathbf{x}_{a,s}$ avec l'ajout de 18 caractéristiques liées à la densité de l'environnement local, portant enfin le nombre total de variables à 223. Ces 18 caractéristiques découlent de la création de 3 nouvelles mesures, calculées pour chacune des 6 sphères de rayons R_k (3 mesures/sphère de rayon R_k $\times$ 6 sphères de rayon R_k = 18). La première est le nombre de résidus se trouvant dans la sphère. La seconde est le nombre de résidus par Å^3. La troisième est la proportion des acides aminés totaux de la protéine se trouvant dans la sphère. À notre connaissance, ces caractéristiques n'ont pas été utilisées dans la littérature à ce jour et nous croyons qu'elles constituent de l'information pertinente afin de caractériser l'accessibilité des sites de phosphorylation.

Pour conclure, nous résumons les 3 sous-groupes de variables qui, ensemble, forment les 223

caractéristiques, notées $x^i_{a,s}$, du vecteur d'entrée $\mathbf{x}_{a,s}$ de chacune des 680 659 observations de l'ensemble de données PELM :

1. La fenêtre séquentielle $SEQ_{a,s} : \{x^1_{a,s}, x^2_{a,s}, ..., x^{13}_{a,s}\}$. Centrée sur le résidu a, elle indique le type de a ainsi que celui des 6 acides aminés en amont et en aval, pour un total de 13 variables discrètes.
2. Les caractéristiques liées aux propriétés physico-chimiques : $\{x^{14}_{a,s}, x^{15}_{a,s}, ..., x^{205}_{a,s}\}$. Elles découlent de l'application des 4 méthodes d'agrégation A_1, A_2, A_3, A_4 sur les 8 propriétés physico-chimiques l des acides aminés présents dans les 6 sphères de rayons R_k, pour un total de $6 \times 8 \times 4 = 192$ caractéristiques.
3. Les variables liées à la densité de l'environnement local : $\{x^{206}_{a,s}, x^{207}_{a,s}, ..., x^{223}_{a,s}\}$. Pour chacune des 6 sphères de rayons R_k, le nombre d'acides aminés dans la sphère, le nombre d'acides aminés par Å^3 et la proportion d'acides aminés totaux de la protéine se retrouvant dans la sphère, résultant donc en $6 \times 3 = 18$ caractéristiques.

3.3 Approche systématique d'apprentissage machine

L'apprentissage machine consiste à exploiter des données pour entraîner des modèles mathématiques capables d'apprendre des relations entre les variables et de réaliser des prédictions sur de nouvelles données. Dans notre contexte, il s'agit donc d'entraîner un modèle mathématique sur les données de l'ensemble PELM, afin de prédire l'état de phosphorylation d'acides aminés. L'objectif final est de concevoir un modèle capable d'effectuer des prédictions précises sur des données qu'il n'a jamais rencontrées auparavant, un concept connu sous le nom de « capacité de généralisation ». Les relations entre les variables sont apprises à l'aide de paramètres, notés θ. Dans un contexte de classification binaire comme le nôtre, on peut exprimer la forme générale d'un modèle de prédiction de la phosphorylation d'acides aminés comme suit :

$$\hat{\boldsymbol{y}} = f(\mathbf{X}, \boldsymbol{\theta}). \tag{3.13}$$

Dans cette équation, f est la fonction qui représente le modèle et $\hat{\boldsymbol{y}}$ correspond au vecteur de prédictions de l'état de phosphorylation des résidus fournis en entrée. La matrice $\mathbf{X}$ contient les vecteurs $\mathbf{x}_{a,s}$ qui caractérisent les résidus a au sein de leur protéine s. Puisque nous disposons de 680 659 observations et que chacune est représentée par son vecteur $\mathbf{x}_{a,s}$, composé de 223 variables, la taille de la matrice $\mathbf{X}$ est de 680 659 $\times$ 223. Les paramètres $\boldsymbol{\theta}$ peuvent être appris en minimisant une fonction d'erreur L. En utilisant cette fonction L, il est possible de

déterminer un ensemble de paramètres optimisés, noté $\boldsymbol{\theta}^*$, minimisant l'erreur :

$$\boldsymbol{\theta}^* = \arg\min_{\theta} \frac{1}{N} \sum_{(a,s)\in\mathcal{D}} L(f(\mathbf{x}_{a,s};\boldsymbol{\theta}), y_{a,s}). \tag{3.14}$$

Dans l'équation ci-haut, $\mathcal{D}$ représente l'ensemble des observations, $f(\mathbf{x}_{a,s};\boldsymbol{\theta})$ et $y_{a,s}$ représentent respectivement la prédiction de l'état de phosphorylation du résidu (a, s) et son état de phosphorylation réel.

La présentation de l'approche systématique d'apprentissage débute avec notre méthode de validation, car il est nécessaire de montrer comment les modèles seront évalués avant leur utilisation. Nous abordons ensuite les méthodes de prétraitement de données. S'ensuit la présentation de notre deuxième contribution, une méthode originale d'échantillonnage de données et le balancement des classes. Nous poursuivons avec les méthodes de sélection de variables, parmi lesquelles se trouve notre troisième et dernière contribution originale. L'approche systématique se conclut par la présentation des 5 types de modèles prédictifs employés dans ce projet.

3.3.1 Validation croisée stratifiée

Dans le cadre du développement d'un modèle d'apprentissage supervisé, il est nécessaire de définir une méthode fiable pour évaluer ses performances, avant de l'utiliser sur de nouvelles données. Dans notre cas, puisque l'ensemble PELM est volumineux et présente un déséquilibre des classes, il est essentiel de le segmenter de façon rigoureuse pour entraîner et tester notre modèle. En effet, retenir simplement 80% des données pour l'entraînement et 20% pour la phase de test n'est pas suffisant, car une seule partition de test risque de ne pas représenter fidèlement la totalité des données. C'est dans cette optique que nous utilisons la validation croisée à K plis. Cette méthode consiste à segmenter les données en K partitions (souvent 5 ou 10), et de manière itérative, utiliser une de ces partitions comme ensemble de test, et le reste des données pour l'entraînement. Cela réduit la variance des résultats et le risque d'avoir entraîné le modèle sur un segment de données non représentatif, menant à une surestimation de sa capacité de généralisation, ou l'inverse.

Afin de diminuer davantage la variance des résultats, nous allons répéter R fois le processus de validation croisée à K plis, en réorganisant les échantillons à chaque répétition. Les données d'entraînement retenues pour chaque répétition découlent de notre méthode originale d'échantillonnage et les données de test sont sélectionnées de manière aléatoire. Nous obtenons ainsi une évaluation plus représentative de la capacité de généralisation du modèle.

Étant donné le déséquilibre des classes dans l'ensemble PELM, il est aussi essentiel de stra-

tifier les données. La stratification veille à ce que les échantillons utilisés lors de la validation croisée conservent une certaine proportion d'observations positives et négatives. Nous évitons ainsi que les ensembles d'entraînement et de test aient des proportions très différentes pour chacune des classes, ce qui peut nuire à la phase d'apprentissage du modèle ou biaiser l'évaluation des performances.

Les modèles prédictifs employés font également usage d'hyperparamètres. Ces derniers varient en fonction du type de modèle prédictif employé et leurs valeurs sont fixées avant la phase d'entraînement. Leur configuration peut avoir une incidence importante sur les performances prédictives et, selon le type de modèle employé, le nombre de configurations possibles peut être élevé. Afin de trouver la combinaison d'hyperparamètres maximisant les performances, nous proposons d'utiliser des ensembles de validation. Ceux-ci interviendront lors d'une première étape indépendante, dans un nombre C de boucles de validation croisée. C représente le nombre exact de configurations possibles d'hyperparamètres, pour le type de modèle prédictif considéré. La configuration d'hyperparamètres fournissant les meilleurs résultats lors de cette première étape sera retenue pour l'étape d'estimation de la capacité de généralisation du modèle.

Ainsi, la validation d'un modèle se déroule en deux temps : la définition de la configuration optimale des hyperparamètres parmi C combinaisons possibles, suivie de l'estimation de la capacité de généralisation via la répétition de R boucles de validation croisée à K plis. Pour notre projet, nous fixons $K = 5$ et $R = 10$. La figure 3.8 offre une visualisation du processus complet.

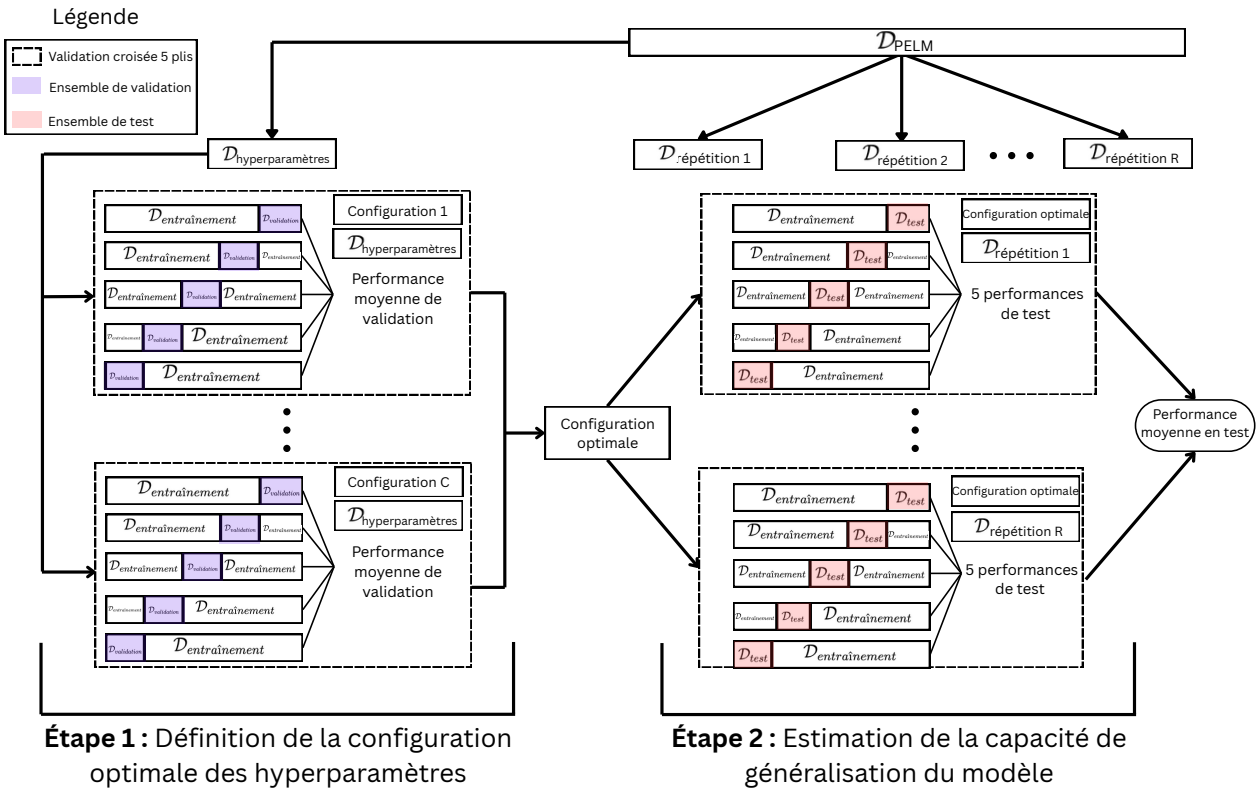

FIGURE 3.8 Processus de validation employé pour notre projet.

3.3.2 Prétraitement des données

L'entraînement de nos modèles est précédé d'une étape de prétraitement des données. Étant donné que nos données comprennent à la fois des variables catégoriques et continues, nous appliquons un encodage aux premières et une normalisation aux secondes. La normalisation est particulièrement importante car elle assure une contribution équitable de chaque variable lors de l'apprentissage, indépendamment de son échelle, évitant ainsi que les variables à grande échelle n'éclipsent les autres. La normalisation des variables permet aussi aux méthodes d'optimisation telles que la descente de gradient stochastique et l'optimiseur Adam de converger plus rapidement. La normalisation permet également à la régression logistique et aux réseaux de neurones de mieux performer, car ces méthodes sont sensibles à l'échelle des données. Nous présentons ainsi 4 méthodes de normalisation et une méthode d'encodage.

Mise à l'échelle maximum absolue

Cette méthode met à l'échelle chaque variable continue en la divisant par sa valeur absolue maximale, plaçant les données dans l'intervalle [–1,1], sans les centrer. Elle préserve ainsi les

zéros, ce qui la rend particulièrement utile pour traiter des données creuses. Dans notre cas, certaines propriétés physico-chimiques (liaisons rotatives, atomes lourds, donneurs/receveurs de pont-H) peuvent être nulles, justifiant la pertinence de considérer cette transformation.

Soit $x^i_{a,s}$, la caractéristique i d'un résidu a au sein d'une protéine s et $\max |x^i|$ la valeur absolue maximale de cette caractéristique (calculée sur l'ensemble d'entraînement), la transformation est donnée par :

$$\tilde{x}^i_{a,s} = \frac{x^i_{a,s}}{\max |x^i|}. \tag{3.15}$$

Mise à l'échelle min-max

Ce type de mise à l'échelle transforme chaque caractéristique continue afin que les valeurs soient comprises dans un intervalle donné (typiquement $[0, 1]$). La forme de la distribution de chaque caractéristique est globalement conservée et l'effet des valeurs aberrantes est atténué. Toutefois, des valeurs près d'être qualifiées d'aberrantes, sans nécessairement l'être, peuvent subir une compression excessive suite à cette transformation. Cela peut donc modifier légèrement la forme des extrémités de la distribution des caractéristiques.

Posons $x^i_{\min}$ la valeur minimale et $x^i_{\max}$ la valeur maximale d'une caractéristique (calculée sur l'ensemble d'entraînement), et $intervalle_{min}$ et $intervalle_{max}$ les bornes de l'intervalle cible. Alors la transformation s'écrit :

$$\tilde{x}'^i_{a,s} = \frac{x^i_{a,s} - x^i_{\min}}{x^i_{\max} - x^i_{\min}} \times (intervalle_{max} - intervalle_{min}) + intervalle_{min}. \tag{3.16}$$

Standardisation

Dans cette approche, la moyenne μ^i d'une caractéristique x^i est soustraite à la valeur de la caractéristique $x^i_{a,s}$ et cette différence est divisée par l'écart-type σ^i. De cette manière, sont centrées et réduites (moyenne nulle et variance unitaire). La transformation s'écrit comme suit :

$$\tilde{x}^i_{a,s} = \frac{x^i_{a,s} - \mu^i}{\sigma^i}. \tag{3.17}$$

Les valeurs aberrantes peuvent toutefois avoir une influence sur le calcul de la moyenne et de l'écart-type. La standardisation n'atténue donc pas complètement l'effet des valeurs aberrantes. Aussi, puisque les données ne sont pas ramenées à un intervalle spécifique, il faut

être conscient que les caractéristiques avec de petites échelles peuvent potentiellement être dominées par des caractéristiques avec de plus grandes échelles lors du processus d'apprentissage d'un modèle. En effet, nos caractéristiques présentent des différences d'échelle non négligeables : par exemple, le coefficient de partage octanol-eau (logP) d'un acide aminé peut varier dans l'intervalle [-4,2 ; -1,1], tandis que la masse moléculaire peut prendre des valeurs comprises entre 75 et 204.

Mise à l'échelle robuste

Cette transformation fait usage de statistiques plus robustes aux données aberrantes, telles que la médiane et l'intervalle interquartile (IQR), plutôt que la moyenne et l'écart-type. En effet, la médiane et l'IQR sont plus robustes car ce sont des méthodes basées sur les rangs. Notons Q_1^i, le premier quartile d'une caractéristique x^i, et Q_3^i son troisième quartile. L'IQR d'une caractéristique x^i, noté IQR^i, est défini par la différence entre Q_3^i et Q_1^i. Ainsi, la mise à l'échelle robuste de la valeur d'une caractéristique x^i d'un acide aminé a au sein d'une protéine s se formule comme suit :

$$\tilde{x}_{a,s}^i = \frac{x_{a,s}^i - \text{médiane}(x^i)}{Q_3^i - Q_1^i}. \tag{3.18}$$

Encodage One-Hot

Utilisé pour les variables catégoriques, l'encodage One-Hot transforme chaque modalité en un vecteur binaire. Dans notre cas, une variable indiquant le type d'un acide aminé à une position donnée d'une fenêtre séquentielle peut prendre l'une des valeurs suivantes : {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, ∅ }. Chacune est alors encodée sous forme vectorielle distincte telle que $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, …, $(0, \ldots, 0, 1)$. Le vecteur binaire a alors une taille de 21, correspondant aux 20 acides aminés possibles et à l'option ∅ pour indiquer l'absence d'acide aminé.

Nous employons également cette méthode d'encodage pour représenter l'état de phosphorylation d'un acide aminé au sein d'une protéine. En effet, la sortie d'un modèle prédictif est un scalaire représentant la probabilité d'appartenir à la classe 1 (représentant un acide aminé phosphorylé), et son complément représente la probabilité d'appartenir à la classe 0. Notre vecteur de prédiction a donc une taille de 2 et est composé de « 1 » et de « 0 », représentant chaque classe.

3.3.3 Échantillonnage et balancement des classes

Pour la phase d'apprentissage, il est essentiel d'utiliser une méthode d'échantillonnage afin de construire des ensembles d'entraînement équilibrés et diversifiés, permettant de maximiser la capacité de généralisation de notre modèle. En effet, il faut éviter de se servir de l'ensemble complet de données pour l'entraînement, en raison du fort déséquilibre entre le nombre de résidus phosphorylés (26 650) et non phosphorylés (654 009). Un tel déséquilibre peut amener le modèle à systématiquement prédire qu'un résidu appartient à la classe négative, car il aurait « appris » durant l'entraînement que cette stratégie maximisait ses performances. Cette forme de surapprentissage mènerait donc à une incapacité à détecter les résidus phosphorylés.

Une approche classique en apprentissage machine pour adresser ce problème consiste à introduire un terme de pénalisation dans la fonction d'erreur, qui dépend de la classe de l'observation pour laquelle l'erreur est calculée. Ainsi, les prédictions erronées effectuées sur des observations appartenant à la classe minoritaire sont pénalisées plus fortement. Cependant, l'incertitude liée à l'appartenance réelle de certains résidus à la classe négative, due à l'absence de validation expérimentale, rend insuffisante ce type d'approche dans notre cas. En effet, cette stratégie pourrait mener à une faible pénalisation dans le cas d'une prédiction erronée sur une observation de la classe négative (classe majoritaire), alors qu'elle appartient en réalité à la classe positive. Dans ce cas précis, une forte pénalisation serait pourtant nécessaire, donc un signal contradictoire serait transmis au modèle, perturbant son apprentissage. Une autre approche consiste à équilibrer directement les classes en sélectionnant, à chaque répétition de la validation croisée, un nouvel échantillon d'observations négatives. Toutefois, en raison du fort déséquilibre initial, il est crucial que ces observations négatives choisies pour l'entraînement soient représentatives de la diversité présente dans l'ensemble complet de données. En effet, entraîner le modèle sur un échantillon de données composé d'observations très semblables risque de limiter sa capacité à généraliser. C'est dans ce contexte que nous proposons une méthode d'échantillonnage permettant d'équilibrer les classes, tout en tenant compte de l'incertitude associée à la classe négative et de l'importance de former des ensembles d'entraînement diversifiés.

Sous-échantillonnage par groupes représentatifs

Cette étape s'inscrit dans le processus de sélection des observations négatives à intégrer au sein des ensembles d'entraînement employés en validation croisée. Les ensembles de test ne sont donc pas soumis à notre méthode de sous-échantillonnage. Les ensembles de test sont plutôt formés aléatoirement et de manière indépendante, afin d'éviter toute forme de surapprentissage.

Nous supposons d'abord qu'il existe, au sein de la classe négative, une certaine proportion de résidus incorrectement classés et nuisibles à la phase d'apprentissage. Cependant, la valeur exacte de cette proportion n'est pas connue et est complexe à estimer. C'est dans cette optique que la première étape de notre méthode consiste à attribuer un score à chaque résidu non phosphorylé, reflétant un niveau de confiance relatif à leur appartenance à la classe négative. Plus le score d'un résidu est haut, plus l'acide aminé correspondant est susceptible d'appartenir en réalité à la classe positive. Les résidus sont ensuite classés selon leur score, et ceux ayant un score au-delà d'un certain percentile (hyperparamètre) ne sont pas éligibles à être inclus dans l'ensemble d'entraînement. La seconde étape de notre méthode consiste à sélectionner, parmi les observations candidates, celles qui seront incluses dans l'ensemble d'entraînement, de sorte à équilibrer les classes. Pour ce faire, les observations sont d'abord projetées dans un espace latent, puis regroupées à l'aide de la méthode des k-moyennes, pour finalement sélectionner un sous-ensemble d'observations au sein de chaque noyau. Les observations retenues parmi chacun des noyaux constituent ce que nous appelons les « groupes représentatifs ».

Pour illustrer la méthode, considérons l'ensemble $\mathcal{D} = \{(\mathbf{x}_{a,s}, y_{a,s}) \mid \mathbf{x}_{a,s} \in \mathbb{R}^{223},\ y_{a,s} \in \{0,1\},\ (a,s) \in \mathcal{P}\}$, où $\mathcal{P}$ désigne toutes les paires (a, s) de résidus a de sérine, de thréonine et de tyrosine au sein des différentes protéines s de PELM. Définissons $\mathcal{D}^+$ et $\mathcal{D}^-$, les sous-ensembles d'observations positives et négatives, respectivement. Puisque nous avons fixé $K = 5$ pour la validation croisée, 5 330 observations positives (20 % de 26 650) sont sélectionnées au hasard parmi $\mathcal{D}^+$ pour former $\mathcal{D}^+_{test}$ et 5 330 observations négatives parmi $\mathcal{D}^-$ sont aussi sélectionnées au hasard pour former $\mathcal{D}^-_{test}$. Ainsi, nous définissons l'ensemble $\mathcal{D}_{test} = \mathcal{D}^-_{test} \cup \mathcal{D}^+_{test}$, qui est mis de côté pour l'instant. Les 21 320 observations positives restantes sont employées pour former $\mathcal{D}^+_{train}$. La stratégie pour sélectionner les 21 320 observations négatives parmi $\mathcal{D}^-_{incertain} = \mathcal{D}^- \backslash \mathcal{D}^-_{test}$ se déroule en 2 phases.

Phase 1 : Réduction de l'incertitude via la construction de $\mathcal{D}^-_{\text{fiable}}$

L'objectif de la première phase est d'identifier les observations, parmi $\mathcal{D}^-_{incertain}$, dont l'appartenance à la classe négative est jugée fiable, afin de constituer l'ensemble $\mathcal{D}^-_{\text{fiable}}$. Pour ce faire, chaque observation se voit d'abord attribuer un score limite, noté $score_{a,s}$, indiquant à quel point $(\mathbf{x}_{a,s}, y_{a,s})$ est susceptible d'appartenir à la classe positive. Le score limite est calculé de la façon suivante :

$$score_{a,s} = \frac{\alpha}{M} \sum_{j=1}^{M} \hat{p}^j_{(a,s)} + \frac{(1-\alpha)}{D_M(\mathbf{x}_{a,s}; \mathcal{D}^+_{train})}. \tag{3.19}$$

Les termes de l'équation 3.19 sont définis comme suit :

— $\hat{p}^j_{a,s}$ représente la probabilité que l'observation $\mathbf{x}_{a,s}$ appartienne à la classe positive, telle que prédite par le j-ème classificateur d'un méta-classificateur constitué de M classificateurs différents.

— $D_M(\mathbf{x}_{a,s}; \mathcal{D}^+_{train})$ est la distance de Mahalanobis entre $\mathbf{x}_{a,s}$ et la distribution des observations d'entraînement positives, $\mathcal{D}^+_{train}$.

— $\alpha \in [0, 1]$ est un hyperparamètre qui contrôle l'équilibre entre l'importance accordée aux prédictions du méta-classificateur et celle attribuée à la distance de Mahalanobis.

La distance de Mahalanobis se calcule comme suit :

$$D_M(\mathbf{x}_{a,s}; \mathcal{D}^+_{train}) = \sqrt{(\mathbf{x}_{a,s} - \boldsymbol{\mu}_{\mathcal{D}^+_{train}})^T \ \boldsymbol{\Sigma}^{-1}_{\mathcal{D}^+_{train}} \ (\mathbf{x}_{a,s} - \boldsymbol{\mu}_{\mathcal{D}^+_{train}})}. \tag{3.20}$$

Cette distance évalue si l'observation négative $\mathbf{x}_{a,s}$ est atypique par rapport aux observations positives d'entraînement. Ainsi, plus la distance est petite, plus l'observation $\mathbf{x}_{a,s}$ « ressemble » aux observations positives, d'où le choix de considérer l'inverse de la distance dans l'équation du score limite. En effet, la distribution de $\mathcal{D}^-_{incertain}$ n'a pas été choisie comme distribution de référence, car la présence d'observations positives au sein de celle-ci aurait biaisé les estimations de $\boldsymbol{\mu}_{\mathcal{D}^-_{incertain}}$ et de $\boldsymbol{\Sigma}^{-1}_{\mathcal{D}^-_{incertain}}$ dès le départ.

La distance de Mahalanobis est favorisée par rapport à d'autres types de distances pour 3 raisons. Premièrement, puisqu'il existe des différences importantes d'échelles entre les caractéristiques, une distance euclidienne risquerait d'être influencée de manière disproportionnée par la ou les variables dont l'échelle est la plus grande. Grâce à la matrice de covariance, chaque caractéristique est normalisée en fonction de sa variance, atténuant cet effet de « dominance » indésirable. Deuxièmement, la matrice de covariance permet de considérer les corrélations entre les variables, ce qui est utile dans un contexte biologique comme le nôtre. En effet, puisque la covariance entre deux variables est définie par $\sigma_{xy} = E[(x - \mu_x)(y - \mu_y)]$ et la corrélation par $\rho_{xy} = \frac{E[(x-\mu_x)(y-\mu_y)]}{(\sigma_x \sigma_y)}$, la corrélation entre ces variables peut s'exprimer en fonction de leur covariance telle que : $\rho_{xy} = \frac{\sigma_{xy}}{(\sigma_x \sigma_y)}$. Troisièmement, la distance de Mahalanobis est plus robuste dans un environnement à haute dimension. En effet, Aggarwal et al. (2001) [51] ont montré que ce type de distance, en tenant compte des covariances, cible les directions de variabilité pertinentes, améliorant ainsi la capacité de distinction entre les observations en haute dimension.

Il est aussi important de noter que nous faisons usage de la matrice inverse $\mathbf{\Sigma}^{-1}_{\mathcal{D}^+_{train}}$, impliquant que la matrice de covariance $\mathbf{\Sigma}_{\mathcal{D}^+_{train}}$ doit d'abord être inversible. Cependant, l'utilisation de l'encodage one-hot, pour encoder les 13 variables discrètes découlant de la fenêtre $SEQ_{a,s}$, rend la matrice de covariance $\mathbf{\Sigma}_{\mathcal{D}^+_{train}}$ singulière. Le théorème et la preuve soutenant cette déclaration se trouvent à l'annexe B. Une matrice singulière ne peut être inversée, car son déterminant est nul. Pour contourner ce problème, nous utilisons la matrice pseudo-inverse de Moore-Penrose, une approximation utilisée lorsqu'une matrice de covariance est singulière et son inverse doit être employée dans le calcul d'une distance. Pour toute matrice réelle A, la matrice pseudo-inverse, A^+, est donnée par :

$$A^+ = VD^+U^*. \tag{3.21}$$

La formulation de A^+ découle directement de la décomposition en valeurs singulières de A : $A = UDV^*$. Dans ces expressions, U et V sont des matrices unitaires, tandis que U^* et V^* désignent leurs matrices adjointes respectives. La matrice D est diagonale, et D^+ s'obtient en transposant D puis en remplaçant chaque valeur diagonale non nulle par son inverse.

Dans l'équation de score, nous retrouvons également les termes $\hat{p}^j_{a,s}$, représentant les probabilités de phosphorylation obtenues par $M = 5$ classificateurs de types différents, appartenant à un méta-classificateur. Les types de modèles utilisés sont les 5 modèles présentés dans la section *Modèles prédictifs*, soit : l'analyse par discriminant linéaire (LDA), la régression logistique, les réseaux de neurones artificiels (RNA), les forêts aléatoires et XGBoost. L'utilisation conjointe de 5 types de classificateurs permet de réduire la variance associée à la moyenne des probabilités $\hat{p}^j_{a,s}$, renforçant ainsi la robustesse du calcul des scores limites. Il aurait été possible d'utiliser un vote binaire pour agréger les prédictions $\hat{p}^j_{a,s}$ du méta-classificateur, mais une moyenne est mieux adaptée, car celle-ci offre plus de granularité.

Pour prévenir le surapprentissage lors du calcul des probabilités $\hat{p}^j_{a,s}$, nous utilisons deux méta-classificateurs entraînés sur des ensembles d'observations distincts. Plus précisément, les ensembles $\mathcal{D}^+_{train}$ et $\mathcal{D}^-_{incertain}$ sont divisés aléatoirement en deux sous-ensembles mutuellement exclusifs, notés $\mathcal{D}_1$ et $\mathcal{D}_2$. Le premier méta-classificateur est entraîné sur $\mathcal{D}_1$ et génère les probabilités $\hat{p}^j_{a,s}$ des observations négatives $(\mathbf{x}_{a,s}; y_{a,s}) \in \mathcal{D}_2$, tandis que le second méta-classificateur suit la procédure inverse. Ainsi, les probabilités $\hat{p}^j_{a,s}$ utilisées pour le calcul des scores limites ne proviennent jamais d'un méta-classificateur ayant déjà vu les mêmes données en entraînement.

Les observations sont ensuite ordonnées selon $percentile_{score_{a,s}}$, le percentile associé à leur

score. Nous définissons ainsi :

$$\mathcal{D}^{-}_{\text{fiable}} = \{(\mathbf{x}_{a,s}, y_{a,s}) \in \mathcal{D}^{-}_{incertain} \mid percentile_{score_{a,s}} < \tau\}. \tag{3.22}$$

Dans cette équation, $\tau \in [0, 1]$ est un hyperparamètre qui permet de contrôler la sévérité selon laquelle une observation est considérée fiable.

Phase 2 : balancement des classes via le sous-échantillonnage de $\mathcal{D}^{-}_{\text{fiable}}$

Il faut maintenant sélectionner $|\mathcal{D}^{+}_{train}| = 21\ 320$ observations parmi $\mathcal{D}^{-}_{\text{fiable}}$, afin de former $\mathcal{D}^{-}_{train}$. Pour ce faire, nous réduisons d'abord la dimension des observations $\mathbf{z}_{a,s} \in \mathcal{D}^{-}_{\text{fiable}}$ à l'aide de 2 auto-encodeurs, pour ensuite regrouper les observations à l'aide de la méthode des k-moyennes. Chaque noyau est ensuite sous-échantillonné de manière à couvrir l'espace des observations le plus efficacement possible en traitant la tâche de sous-échantillonnage comme un problème k-centre.

Les détails concernant la structure précise des auto-encodeurs utilisés pour la réduction dimensionnelle sont fournis dans la section *Sélection de variables* du mémoire. Dans la présente section, l'attention est plutôt portée sur les raisons qui nous ont menés à intégrer cette étape à notre méthode d'échantillonnage. Cette réduction dimensionnelle atténue les effets négatifs du « fléau de la dimensionnalité » lors du calcul de distances en haute dimension et assure également la continuité des variables utilisées dans ces mêmes calculs de distances. En effet, l'algorithme standard des k-moyennes fait usage d'une distance euclidienne entre deux observations $\mathbf{x}_{(a,s)_i}$ et $\mathbf{x}_{(a,s)_j}$ définie comme suit :

$$d(\mathbf{x}_{(a,s)_i}; \mathbf{x}_{(a,s)_j}) = \sqrt{\sum_{k=1}^{d} (x^{(k)}_{(a,s)_i} - x^{(k)}_{(a,s)_j})^2}. \tag{3.23}$$

Cependant, au fur et à mesure que le nombre de dimensions d augmente, la distance peut perdre de sa signification si de nombreuses caractéristiques comportent du bruit ou sont redondantes. C'est donc ce qui nous motive à d'abord employer un espace latent pour représenter les observations. Ainsi, dans notre version de l'algorithme des k-moyennes, la distance euclidienne est remplacée par :

$$d(\mathbf{z}_{(a,s)_i}; \mathbf{z}_{(a,s)_j}) = \sqrt{\sum_{k=1}^{d_{latent}} (z^{(k)}_{(a,s)_i} - z^{(k)}_{(a,s)_j})^2}, \tag{3.24}$$

où les vecteurs $\mathbf{z}_{(a,s)_i}$ et $\mathbf{z}_{(a,s)_j}$ représentent les observations $\mathbf{x}_{(a,s)_i}$ et $\mathbf{x}_{(a,s)_j}$ dans l'espace latent de dimension d_{latent}.

L'algorithme des k-moyennes crée ensuite un ensemble de k noyaux, noté $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$. Pour ce faire, la distance au carré intra-noyau est minimisée comme suit :

$$\arg\min_{\mathcal{C}} \sum_{i=1}^{k} \sum_{\mathbf{z}_{a,s} \in \mathcal{C}_i} \|\mathbf{z}_{a,s} - \boldsymbol{\mu}_i\|^2 \;=\; \arg\min_{\mathcal{C}} \sum_{i=1}^{k} |\mathcal{C}_i| \operatorname{Var}(\mathcal{C}_i). \tag{3.25}$$

Dans cette équation, $\boldsymbol{\mu}_i$ représente le centroïde du noyau $\mathcal{C}_i$ et est défini comme suit : $\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{z}_{a,s} \in \mathcal{C}_i} \mathbf{z}_{a,s}$. Des valeurs de k comprises entre 2 et 20 ont été évaluées, et la valeur optimale, maximisant les performances prédictives à l'étape de sous-échantillonnage, s'est révélée être $k = 8$.

Il faut maintenant réduire la taille de chacun des noyaux $\mathcal{C}_i$, afin d'obtenir k ensembles représentatifs, notés $\mathcal{C}_{i,groupe}$. Un ensemble représentatif doit refléter le plus fidèlement possible la diversité des observations qui constituent son noyau $\mathcal{C}_i$ correspondant. Pour ce faire, nous traitons chaque noyau $\mathcal{C}_i$, muni de la distance euclidienne entre ses observations, comme un espace métrique dont la taille est réduite en résolvant le problème de k-centre. Ce processus est répété indépendamment pour chaque noyau $\mathcal{C}_i$. En théorie des graphes, le problème de k-centre consiste à trouver un ensemble de sommets pour lequel la plus grande distance d'un point à son sommet le plus proche est minimale. Formellement, définissons un espace métrique $(\mathcal{C}_i, L2)$, où $\mathcal{C}_i$ est un des k noyaux découlant de l'étape de regroupement par k-moyennes précédente et L2 la distance euclidienne entre les observations $\mathbf{z}_{a,s} \in \mathcal{C}_i$. La preuve que $(\mathcal{C}_i, L2)$ satisfait la définition d'un espace métrique $\forall\ \mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k\}$ se trouve à l'annexe C. L'objectif est de déterminer un sous-ensemble d'observations $\mathcal{C}_{i,groupe} \subseteq \mathcal{C}_i : |\mathcal{C}_{i,groupe}| = v_i$, tel que $\mathcal{C}_i$ puisse être entièrement couvert par v_i sphères de rayon minimal r_i, chacune étant centrée sur une observation $\mathbf{z}_{a,s}$. Ainsi, la fonction de coût à minimiser se formule comme suit :

$$r_i \;=\; \max_{\mathbf{z} \in \mathcal{C}_i} \; d\Big(\mathbf{z}; \mathcal{C}_{i,ensemble}\Big). \tag{3.26}$$

Dans cette équation, $d\Big(\mathbf{z}; \mathcal{C}_{i,groupe}\Big)$ représente l'ensemble des distances entre tous les points $\mathbf{z}_{a,s} \in \mathcal{C}_i$ et les observations du groupe représentatif $\mathcal{C}_{i,groupe}$. Ainsi, quelconque point défini dans $\mathbb{R}^{d_{latent}}$ d'un noyau $\mathcal{C}_i$ est garanti d'être à une distance maximale r_i d'une observation $\mathbf{z}_{a,s} \in \mathcal{C}_{i,groupe}$. Nous définissons le nombre d'observations v_i à sélectionner parmi chaque noyau $\mathcal{C}_i$ comme suit :

$$v_i = |\mathcal{C}_{i,groupe}| = \frac{|\ \mathcal{C}_i\ |}{\sum_i^k |\ \mathcal{C}_i\ |} \cdot |\ \mathcal{D}^+_{train}\ |\ \ \forall\ i = \{1, 2, .., k\}. \tag{3.27}$$

Puisque les valeurs $|\mathcal{D}^+_{train}|$ (80% de $|\mathcal{D}^+|$) et $|\mathcal{C}_i|\ \forall\ k \in \{1, 2, ..., k\}$ sont déjà connues à

cette étape-ci, il suffit de les substituer dans l'équation 3.27. Le problème de k-centre est un problème classé NP-difficile, signifiant qu'il n'existe pas d'algorithme efficace garantissant la solution optimale. Nous utilisons ainsi un algorithme développé par Gonzal et al. (1985) [52], qui permet de trouver une solution en un temps raisonnable, pour laquelle le rayon r_i trouvé est garanti d'être plus petit ou égal à 2 fois celui de la solution optimale. L'algorithme est le suivant :

Algorithm 1 Algorithme « Farthest-first traversal »

Require: Ensemble $\mathcal{C}_i$, entier v_i

1: Choisir aléatoirement un point $\mathbf{z_j} \in \mathcal{C}_\text{i}$
2: $\mathcal{C}_{i,groupe} \leftarrow \mathbf{z_j}$
3: **while** $|\mathcal{C}_{i,groupe}| < v_i$ **do**
4: $\quad \mathbf{z_j} \leftarrow \arg\max_{\mathbf{z_k} \in \mathcal{C}_i} L2\Big(\mathbf{z_k}, \mathcal{C}_{i,groupe}\Big)$
5: $\quad \mathcal{C}_{i,groupe} \leftarrow \mathcal{C}_{i,groupe} \cup \{\mathbf{z_j}\}$
6: **end while**

L'union des groupes représentatifs forme ainsi l'ensemble $\mathcal{D}^-_{train}$, tel que : $\mathcal{D}^-_\text{train} = \bigcup_{i=1}^{k} \mathcal{C}_{i,\text{groupe}}$. Le processus complet de la construction des groupes représentatifs, à partir de l'ensemble $\mathcal{D}^-_{fiable}$ est illustré à la figure 3.9.

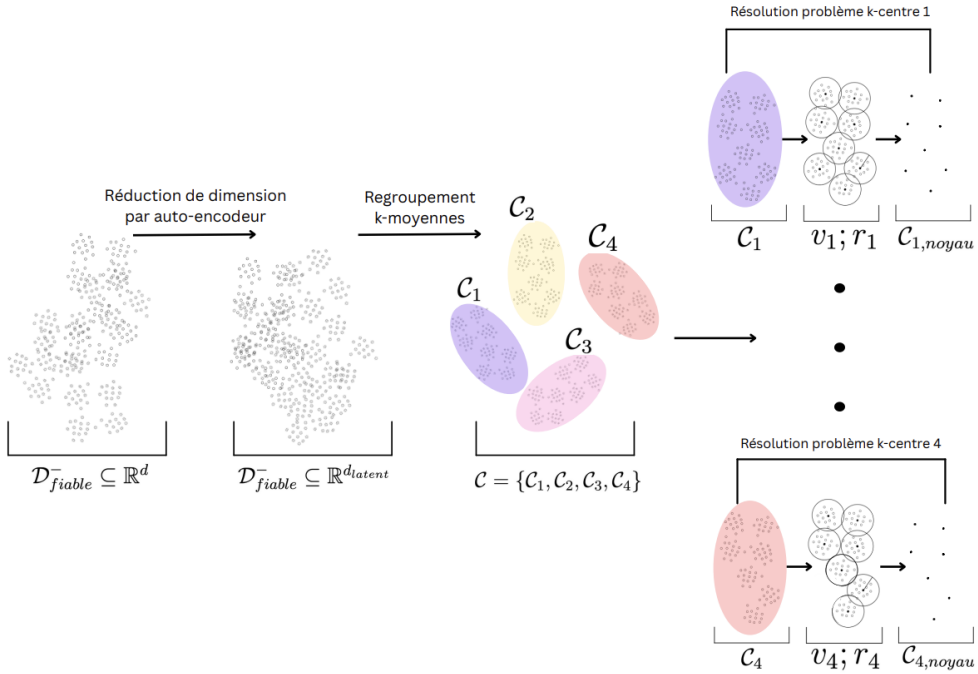

FIGURE 3.9 Illustration d'un exemple simplifié de la construction de groupes représentatifs à partir de $\mathcal{D}^-_{fiable}$, en fixant $k = 4$.

Algorithm 2 Algorithme de sous-échantillonnage par groupes représentatifs

Require: $\mathcal{D}^+_{\text{train}}, \mathcal{D}^-, \alpha, \tau, k, d_{\text{latent}}$
$\mathcal{D}^-_{\text{incertain}} \leftarrow \mathcal{D}^- \setminus \mathcal{D}^-_{\text{test}}$
for all $\mathbf{x}_{a,s} \in \mathcal{D}^-_{incertain}$ **do**
    $score_{a,s} \leftarrow \frac{\alpha}{M} \sum_{j=1}^{M} \hat{p}^j_{a,s} \; + \; \frac{1-\alpha}{D_M(\mathbf{x}_{a,s};\, \mathcal{D}^+_{\text{train}})}$
    if $score_{a,s} \leq \tau$ **then**
        $\mathcal{D}^-_{\text{fiable}} \leftarrow \mathcal{D}^-_{\text{fiable}} \cup \{\mathbf{x}_{a,s}\}$
    end if
end for
$Z \leftarrow \text{autoencodeur}(\mathcal{D}^-_{\text{fiable}}; d_{latent})$
$\{C_1, \ldots, C_k\} \leftarrow \text{k-moyenne}(Z,\, k)$
for $i = 1$ **to** k **do**
    $\mathcal{C}_{i,\text{groupe}} \leftarrow \text{k-centre}(C_i,\, v_i)$
end for
$\mathcal{D}^-_{\text{train}} \leftarrow \bigcup_{i=1}^{k} \mathcal{C}_{i,\text{groupe}}$
$\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}^-_{\text{train}} \; \cup \; \mathcal{D}^+_{\text{train}}$

3.3.4 Sélection de variables

La sélection de variables peut être séparée en deux grandes familles : les techniques de sélection et les techniques de réduction de dimension. La sélection consiste à identifier la meilleure combinaison de variables, selon un ou des critères fixés. La réduction de dimension consiste plutôt à construire un espace latent qui synthétise l'information en exploitant les relations existantes entre les variables initiales. Cette section porte principalement sur les méthodes de sélection, car contrairement à la réduction de dimension, les variables ne sont pas représentées dans un espace latent, permettant ainsi de conserver l'interprétation des résultats. Cette section débute avec la présentation des techniques de sélection, parmi lesquelles se retrouve une approche originale développée spécifiquement pour notre problématique. S'ensuit l'unique méthode de réduction employée dans ce projet, la réduction de dimension par auto-encodeur (employée dans le cadre de notre méthode d'échantillonnage).

Méthodes de sélection

Trois types de méthodes de sélection seront utilisées dans notre étude : les méthodes de filtre, les méthodes intégrées et les méthodes enveloppantes (« wrapper » en anglais). Les méthodes de filtre sélectionnent les variables en fonction d'indicateurs ou de tests statistiques, indépendamment des métriques de performance de classification. Les méthodes intégrées effectuent la sélection directement dans le cadre du processus d'apprentissage du modèle. Les méthodes enveloppantes, quant à elles, recherchent la combinaison optimale de variables en

fonction d'une métrique de performance fixée pour la tâche à accomplir.

L'information mutuelle, définie comme suit :

$$I(\mathbf{x}_i;\mathbf{x}_j) = \iint p(x_i,x_j)\ \log\frac{p(x_i,x_j)}{p(x_i)\,p(x_j)}dx \tag{3.28}$$

et la corrélation de Spearman :

$$\rho(\mathbf{x}_i,\mathbf{x}_j) = \frac{\sum_{n=1}^{N}\Big(R_i^{(n)}-\overline{R_i}\Big)\Big(R_j^{(n)}-\overline{R_j}\Big)}{\sqrt{\sum_{n=1}^{N}\Big(R_i^{(n)}-\overline{R_i}\Big)^2}\sqrt{\sum_{n=1}^{N}\Big(R_j^{(n)}-\overline{R_j}\Big)^2}}, \tag{3.29}$$

sont deux critères de type filtre qui seront employés pour mesurer la redondance entre les caractéristiques. Pour un couple de variables, plus ces deux mesures sont élevées, plus les variables sont corrélées entre elles. On peut ainsi identifier et retirer les variables trop corrélées. Nous utilisons également le test non paramétrique de Mann-Whitney U pour détecter les variables qui ont des distributions différentes, en séparant les données selon leur classe. Ainsi, pour une variable x_i, le test de Mann–Whitney U vérifie les hypothèses suivantes :

$H_0 : F_{\mathbf{x}_i^+} = F_{\mathbf{x}_i^-}$ (distributions identiques) contre $H_1 : F_{\mathbf{x}_i^+} \neq F_{\mathbf{x}_i^-}$ (distributions différentes).

En effectuant ce test pour toutes les variables, nous pouvons identifier celles qui ne sont pas pertinentes vis-à-vis la variable cible et les retirer.

Nous allons également utiliser deux critères de redondance minimale et pertinence maximale (mRMR). Un critère mRMR classe les caractéristiques en fonction d'une combinaison de la pertinence et de la redondance. Ainsi, la méthode mRMR réunit dans un même score les concepts de pertinence et de redondance, contrairement à utiliser de façon individuelle des tests statistiques pour la pertinence vis-à-vis la variable cible et des tests de corrélation pour la redondance entre les caractéristiques. Les 2 critères mRMR qui seront utilisés sont les suivants :

$$\text{Critère mRMR 1 : } \max_{\mathbf{x}_i\in\mathbf{A}-\mathbf{S}}\Big(F(\mathbf{x}_i,y)\ -\ \frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}_j\in\mathbf{S}}\big|\rho(\mathbf{x}_i,\mathbf{x}_j)\big|\Big), \tag{3.30}$$

$$\text{Critère mRMR 2 : } \max_{\mathbf{x}_i\in\mathbf{A}-\mathbf{S}}\left(I(\mathbf{x}_i;y)-\frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}_j\in\mathbf{S}}I(\mathbf{x}_i;\mathbf{x}_j)\right) \tag{3.31}$$

Dans ces équations, les termes $F(\mathbf{x}_i, y)$ et $I(\mathbf{x}_i; y)$ mesurent la pertinence de $\mathbf{x}_i$ via ANOVA/F et l'information mutuelle, respectivement. D'autre part, les termes $\frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}_j \in \mathbf{S}} \left|\rho(\mathbf{x}_i, \mathbf{x}_j)\right|$ et $\frac{1}{|\mathbf{S}|}\sum_{\mathbf{x}_j \in \mathbf{S}} I(\mathbf{x}_i; \mathbf{x}_j)$ mesurent la redondance entre les caractéristiques, via la corrélation de Spearman et l'information mutuelle. $\mathbf{A}$ est l'ensemble total de variables et $\mathbf{S}$ est le sous-ensemble de variables déjà choisies. Nous anticipons que l'application du critère 2 mRMR conduira à de meilleurs résultats, car l'information mutuelle repose sur moins d'hypothèses concernant les données que la corrélation de Spearman (relation monotone entre les variables) et l'ANOVA/F (normalité).

Les méthodes intégrées employées comprennent la régularisation Lasso et les forêts aléatoires, toutes deux présentées en détail dans la section consacrée aux modèles prédictifs. La régularisation Lasso est introduite comme technique de régularisation et est directement intégrée aux fonctions de perte des modèles prédictifs. Cette approche ramène à zéro les coefficients des variables qui ne contribuent pas à réduire l'erreur pendant l'entraînement. D'autre part, les forêts aléatoires peuvent être adaptées à la sélection de variables et constituent une méthode pertinente dans notre contexte, car elles peuvent capturer des relations non linéaires entre les variables et offrent de bonnes performances en haute dimension. Une sous-section distincte est consacrée aux forêts aléatoires, puisqu'elles figurent également parmi les cinq modèles prédictifs retenus pour la tâche de prédiction.

Les méthodes enveloppantes sont particulièrement intéressantes puisqu'elles évaluent directement la pertinence des sous-ensembles en fonction de la performance du modèle, permettant de capturer des relations complexes et non linéaires entre les caractéristiques et la variable cible. Les méthodes enveloppantes classiques incluent la sélection ascendante, dans laquelle le point de départ est le sous-ensemble vide et on ajoute progressivement les variables qui contribuent le plus à augmenter les performances. Une autre méthode est la sélection arrière, qui a plutôt comme point de départ l'ensemble complet de variables, et qui, progressivement, retire les variables les moins pertinentes. Ces méthodes sont simples et efficaces, mais elles peuvent rapidement devenir très coûteuses en termes computationnels. Plus spécifiquement, dans notre cas, évaluer exhaustivement toutes les combinaisons possibles des 223 caractéristiques contenues dans les vecteurs $\mathbf{x}_{a,s}$ impliquerait d'entraîner et de tester 2^{223} modèles différents, ce qui est impossible en pratique.

Pour contourner ce problème, il est possible d'utiliser des heuristiques. Le recuit simulé, les algorithmes génétiques et l'essaim de particules sont trois exemples d'heuristiques couramment utilisées. Ces approches permettent d'explorer l'espace de solutions plus intelligemment, diminuant le coût de calcul, tout en obtenant des sous-ensembles de variables quasi-optimaux. Ces heuristiques sont flexibles car elles regroupent plusieurs concepts pouvant être combinés

et adaptés à des applications spécifiques.

Méthode originale de sélection de variables

Nous proposons une méthode originale de sélection de variables adoptant une approche hybride, qui fait appel à des concepts du recuit simulé (méthode enveloppante) ainsi que des tests statistiques et de corrélation (méthodes de filtrage). La méthode génère itérativement des sous-ensembles de variables candidates en sélectionnant de manière probabiliste des caractéristiques au sein de groupes de variables, en fonction de leur pertinence statistique a priori et d'un paramètre de température décroissant. L'influence de la pertinence statistique sur la sélection augmente progressivement à mesure que la température diminue, guidant l'algorithme d'une phase d'exploration vers une phase d'exploitation. À chaque itération, chaque solution candidate est évaluée selon sa performance prédictive et, si la performance est supérieure celle de la solution actuelle, celle-ci est remplacée par la solution candidate.

Phase 1 : Création du sous-ensemble de solutions $\mathcal{S}$

Considérons l'ensemble $\mathcal{A}$, qui contient l'ensemble des combinaisons possibles des 223 variables employées dans notre modélisation, soit 2^{223} solutions uniques. La phase 1 de notre méthode consiste à extraire de $\mathcal{A}$ un sous-ensemble restreint, noté $\mathcal{S}$, dans lequel notre algorithme peut puiser une solution à chaque itération k. Considérons d'abord les sous-groupes de variables suivants :

- SEQ : la fenêtre séquentielle de 13 variables indiquant le type des acides aminés voisins.
- PC_1 à PC_8 : 8 groupes contenant chacun 24 variables issues des 4 méthodes d'agrégation appliquées aux 6 sphères de rayons distincts pour chacune des 8 propriétés physico-chimiques.
- D : un groupe de 18 variables décrivant la densité de l'environnement local, calculées à partir de trois mesures de densité calculées pour les 6 sphères.

Notons S_{k+1}, une solution candidate de combinaison de variables, construite par notre algorithme à l'itération k. Les 13 variables de SEQ sont automatiquement incluses dans chacune des solutions candidates S_{k+1}. Ce choix découle du fait qu'il est souhaitable de conserver la visibilité sur les types d'acides aminés voisins. La lecture des motifs des résidus adjacents nous permet d'inférer sur les kinases liées aux événements de phosphorylation, ce qui est utile pour identifier des biomarqueurs. En effet, le dérèglement de l'activité chimique de certaines

familles de kinases est associé à l'hyperphosphorylation de plusieurs protéines, phénomène observé chez les patients atteints de maladies neurodégénératives [53]. Conserver SEQ revient donc à garder la capacité de repérer en priorité les événements de phosphorylation induits par des kinases liées aux maladies neurodégénératives.

Pour les groupes PC_1 à PC_8 et D, nous définissons deux variables discrètes aléatoires, t_{k+1} et q_{k+1}, qui définissent le nombre de variables à sélectionner parmi chacun des groupes, respectivement. Ainsi, à chaque itération k de l'algorithme, la variable aléatoire discrète t_{k+1} dicte le nombre de variables à sélectionner au sein des sous-groupes PC_1 à PC_8 et est définie comme suit :

$$t_{k+1} \sim \text{Unif}\Big\{1, \ldots, \mathcal{T}\Big\}, 1 \leq \mathcal{T} \leq 24. \tag{3.32}$$

L'entier $\mathcal{T}$ est un hyperparamètre de la méthode et se situe entre 1 et 24, car $|PC_i| = 24 \ \forall \ i \in \{1, 2, ..., 8\}$. De la même façon, nous définissons q_{k+1}, qui établit le nombre de variables à sélectionner parmi D, comme suit :

$$q_{k+1} \sim \text{Unif}\Big\{1, \ldots, \mathcal{Q}\Big\}, 1 \leq \mathcal{Q} \leq 18. \tag{3.33}$$

L'entier $\mathcal{Q}$ est également un hyperparamètre et se situe entre 1 et 18, car $|D| = 18$. Ainsi, le nombre maximal de variables incluses dans chaque solution S_{k+1} peut être borné par l'utilisateur, donnant un contrôle direct sur le niveau d'interprétabilité.

Le regroupement du reste des caractéristiques à l'aide des sous-groupes PC_1 à PC_8 et D, ainsi que le choix de conserver au moins une caractéristique parmi ces groupes s'explique à l'aide de 2 arguments. Premièrement, après avoir calculé les corrélations de Spearman entre les variables, nous avons remarqué que les variables découlant de l'agrégation d'une même propriété physico-chimique étaient fortement corrélées entre elles, un résultat attendu. Le même comportement a été observé entre les variables liées à la densité. Cela suggère donc que le nombre de variables retenues parmi chacun de ces sous-groupes pourrait être réduit. Les matrices de corrélation liées à cette analyse sont disponibles en annexe.

Deuxièmement, afin d'évaluer leur potentiel discriminatif, nous avons effectué les 4 tests statistiques suivants pour chacune des variables au sein des sous-groupes PC_1 à PC_8 et D :

1. Test Mann-Whitney U : $H_0 : x_i^+$ et x_i^- partagent la même distribution contre $H_1 : x_i^+$ et x_i^- proviennent de deux distributions différentes. Ici, x_i^+ représente les valeurs de la caractéristique x_i des observations de la classe positive et vice-versa.
2. Test-t de Welch : $H_0 : \mu_{x_i^+} = \mu_{x_i^-}$ contre $H_1 : \mu_{x_i^+} \neq \mu_{x_i^-}$.

3. Test de Kolmogorov-Smirnov : $H_0 : x_i^+$ et x_i^- partagent la même fonction de répartition contre $H_1 : x_i^+$ et x_i^- ont des fonctions de répartition différentes.
4. Test de Levene : $H_0 : \text{var}\left(x_i^+\right) = \text{var}\left(x_i^-\right)$ contre $H_1 : \text{var}\left(x_i^+\right) \neq \text{var}\left(x_i^-\right)$.

Pour chacun des 9 sous-groupes de variables (PC_1 à PC_8 et D), au moins une des variables a démontré un résultat statistiquement significatif lors des tests. Ces tests non paramétriques ont été employés car nous avons constaté que les caractéristiques ne sont pas distribuées normalement, et ces tests ne reposent pas sur cette hypothèse. Nous jugeons donc pertinent d'inclure automatiquement au moins une caractéristique par propriété physico-chimique et au moins une variable liée à la densité. Le choix des variables à inclure sera guidé par notre stratégie d'exploration, décrite à la phase 2 de notre méthode. En effet, il est possible que des variables considérées individuellement redondantes ou peu discriminantes soient pertinentes lorsqu'elles interagissent avec d'autres caractéristiques ou vice-versa. Il est donc nécessaire de disposer d'un mécanisme pour l'exploration systématique de ces combinaisons, afin de ne pas écarter des interactions entre les variables pouvant augmenter la performance du modèle.

Phase 2 : exploration de $\mathcal{S}$

Nous présentons ici la stratégie d'exploration employée pour générer les solutions candidates S_{k+1}. Chaque solution candidate est définie comme l'union des variables de la fenêtre séquentielle SEQ et des variables sélectionnées parmi les autres groupes :

$$S_{k+1} = S_{SEQ} \cup \bigcup_{i=1}^{8} S_{PC_i;k+1} \cup S_{D;k+1}. \tag{3.34}$$

Ici, S_{SEQ} représente les 13 variables fixes du groupe SEQ. Les ensembles $S_{PC_i,k+1}$ et $S_{D,k+1}$ contiennent les variables sélectionnées à l'itération k parmi les groupes PC_i (pour $i = 1, \ldots, 8$) et D, respectivement. Le nombre de variables à sélectionner dans ces groupes (t_{k+1} pour chaque PC_i et q_{k+1} pour D) est déterminé aléatoirement comme décrit dans la Phase 1.

À chaque itération, la sélection des variables qui composent les ensembles $S_{PC_i,k+1}$ et $S_{D,k+1}$ est réalisée de manière probabiliste. Cette sélection est guidée par 9 distributions de probabilités discrètes, $\mathcal{P}_{l,k}$, où l'indice $l \in \{1, \ldots, 9\}$ désigne le groupe de variables concerné. Plus précisément, nous associons les distributions $\mathcal{P}_{1,k}, \ldots, \mathcal{P}_{8,k}$ aux groupes $PC_1, \ldots, PC_8$ respectivement, et la distribution $\mathcal{P}_{9,k}$ au groupe D.

Chaque distribution $\mathcal{P}_{l,k}$ attribue une probabilité de sélection $p_{l,j,k}$ à chaque variable j appartenant au groupe l. Pour définir cette probabilité, considérons d'abord la taille J_l de chaque

groupe l :

$$J_l = \begin{cases} 24 & \text{si } l \in \{1, \ldots, 8\} \quad (\text{groupes } PC_1, \ldots, PC_8) \\ 18 & \text{si } l = 9 \quad (\text{groupe } D) \end{cases} \tag{3.35}$$

La probabilité $p_{l,j,k}$ pour la variable j du groupe l (où $1 \leq j \leq J_l$) à l'itération k est ensuite calculée via une fonction softmax, pondérée par un score de pertinence et un paramètre de température T_k :

$$p_{l,j,k} = \frac{\exp\Big(\text{pertinence}_{l,j}/T_k\Big)}{\sum_{j'=1}^{J_l} \exp\Big(\text{pertinence}_{l,j'}/T_k\Big)}. \tag{3.36}$$

Le terme $\text{pertinence}_{l,j}$ est un score quantifiant le pouvoir discriminatif a priori de la variable j du groupe l. Formellement, ce score est défini pour une variable par la somme des logarithmes négatifs des p-values obtenues sur l'ensemble $\mathcal{E}$ des 4 tests statistiques employés lors de la phase 1 (Mann-Whitney U, Welch t-test, Kolmogorov-Smirnov, Levene) :

$$\text{pertinence}_{l,j} = -2 \sum_{\epsilon \in \mathcal{E}} \ln \text{p-valeur}_{l,j,\epsilon}. \tag{3.37}$$

Ici, $\text{p-valeur}_{l,j,\epsilon}$ désigne la p-valeur associée à la variable j du groupe l pour le test statistique ϵ. La formule retenue pour la pertinence reprend le procédé de combinaison de p-values proposé par Fisher [54]. Il a montré que le résultat de cette somme suit une loi du chi-carré, avec 2ϵ degrés de liberté, permettant d'utiliser cette statistique pour fusionner les résultats de plusieurs tests. Ainsi, une variable qui demeure fortement significative dans l'ensemble des tests obtient un score de pertinence élevé, tandis qu'une variable peu ou pas significative reçoit un score plus faible.

La température T_k est un hyperparamètre modulant l'influence du score de pertinence sur les probabilités de sélection. Lorsque la température T_k est élevée, les différences relatives entre les scores de pertinence sont atténuées et les probabilités $p_{l,j,k}$ auront tendance à s'uniformiser. Cela permet à l'algorithme de considérer des variables jugées moins pertinentes a priori, mais qui pourraient se révéler utiles en interaction avec d'autres. Inversement, à mesure que T_k diminue au fil des itérations, l'influence du score de pertinence est amplifiée : les variables ayant une pertinence élevée se voient attribuer une probabilité de sélection plus grande.

La température T_k diminue progressivement au fil des itérations de l'algorithme, suivant un processus de refroidissement contrôlé par plusieurs hyperparamètres. Ces hyperparamètres incluent la température initiale T_0, qui fixe le niveau d'exploration au démarrage ; le taux

de refroidissement $\alpha \in [0, 1]$, un facteur multiplicatif qui détermine la vitesse de diminution de la température ; le nombre d'itérations M à effectuer pour chaque palier de température ; et la température minimale T_{min}, qui sert de critère d'arrêt à l'algorithme. Après chaque bloc de M itérations pendant lesquelles des solutions candidates sont générées et évaluées, la température est mise à jour comme suit :

$$T_{k+1} = \alpha \times T_k. \tag{3.38}$$

Ainsi, à chaque itération k, l'algorithme évalue la solution candidate S_{k+1}. Si la performance de la solution S_{k+1} est supérieure à la performance obtenue avec la solution actuelle S_k, S_{k+1} devient la solution actuelle.

Algorithm 3 Méthode Originale de Sélection de Variables

Require:

Ensembles de variables V_{SEQ}, $\{V_{PC_l}\}_{l=1}^{8}$, V_D
Scores pertinence$_{l,j}$ pour $j \in V_l$, $l \in \{1, \ldots, 9\}$
Hyperparamètres $\mathcal{T}$, $\mathcal{Q}$, T_0, α, M, T_{min}
Initialisation : $T \leftarrow T_0$, $k \leftarrow 0$, $S_{current} \leftarrow$ GénérerSolutionInitiale(), $Perf_{current} \leftarrow Perf(S_{current})$, $S_{best} \leftarrow S_{current}$, $Perf_{best} \leftarrow Perf_{current}$
Boucle Principale :
while $T > T_{min}$ **do**
    for $m = 1$ to M **do**
        Tirer $t \sim \text{Unif}\{1, \ldots, \mathcal{T}\}$, $q \sim \text{Unif}\{1, \ldots, \mathcal{Q}\}$
        $S_{candidate} \leftarrow V_{SEQ}$
        for $l = 1$ to 8 **do**
            Calculer $p_{l,j,k} \propto \exp(\text{pertinence}_{l,j}/T)\ \forall j \in V_{PC_l}$
            $S_{candidate} \leftarrow S_{candidate} \cup$ Échantillonner$(V_{PC_l}, t, \{p_{l,j,k}\})$
        end for
        Calculer $p_{9,j,k} \propto \exp(\text{pertinence}_{9,j}/T)\ \forall j \in V_D$
        $S_{candidate} \leftarrow S_{candidate} \cup$ Échantillonner$(V_D, q, \{p_{9,j,k}\})$
        $Perf_{candidate} \leftarrow Perf(S_{candidate})$
        if $Perf_{candidate} > Perf_{current}$ **then**
            $S_{current} \leftarrow S_{candidate}$, $Perf_{current} \leftarrow Perf_{candidate}$
            if $Perf_{current} > Perf_{best}$ **then**
                $S_{best} \leftarrow S_{current}$, $Perf_{best} \leftarrow Perf_{current}$
            end if
        end if
        $k \leftarrow k + 1$
    end for
    $T \leftarrow \alpha \times T$
end while
return S_{best}

Réduction de dimension par auto-encodeur

Un auto-encodeur est un type de modèle qui exploite les réseaux de neurones artificiels et qui peut être adapté pour diverses tâches, dont la réduction de dimension. L'architecture de ce genre de modèle comporte 3 éléments : un encodeur, un espace latent et un décodeur.

L'encodeur est un réseau neuronal profond de L couches cachées où le nombre de neurones de chacune des couches est graduellement réduit, jusqu'à atteindre le niveau de compression désiré, correspondant à la dimensionnalité de l'espace latent. Le décodeur est aussi un réseau de neurones artificiels profond ayant L couches cachées, mais procède plutôt à une augmentation graduelle du nombre de neurones à travers les couches cachées pour atteindre un nombre de neurones équivalent à la dimensionalité des observations dans l'espace original. Dans notre cas, nous utilisons les auto-encodeurs uniquement lors de notre étape d'échantillonnage et ceux-ci manipulent exclusivement des observations négatives appartenant à $\mathcal{D}^{-}_{fiable}$.

À partir d'un vecteur d'entrée $\mathbf{x}_{a,s} \in \mathbb{R}^d$, l'encodeur applique la transformation suivante :

$$q_{\boldsymbol{\phi}} : \mathbb{R}^d \longrightarrow \mathbb{R}^{d_{\text{latent}}}, \qquad \mathbf{z}_{a,s} = q_{\boldsymbol{\phi}}(\mathbf{x}_{a,s}). \tag{3.39}$$

Dans l'expression ci-haut, $\mathbf{z}_{a,s} \in \mathbb{R}^{d_{\text{latent}}}$ représente l'entrée $\mathbf{x}_{a,s}$ dans l'espace latent, $d_{\text{latent}} \ll d$, et $\boldsymbol{\phi}$ rassemble les poids et biais du réseau encodeur. Les observations $\mathbf{z}_{a,s}$ sont ensuite fournies en entrée au décodeur, qui est paramétré par $\boldsymbol{\theta}$. Le rôle de l'encodeur est de reconstruire les observations $\mathbf{z}_{a,s}$ comme suit :

$$p_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\text{latent}}} \longrightarrow \mathbb{R}^d, \qquad \widehat{\mathbf{x}}_{a,s} = p_{\boldsymbol{\theta}}(\mathbf{z}_{a,s}). \tag{3.40}$$

L'objectif de la phase d'entraînement consiste à minimiser l'erreur de reconstruction, définie par l'erreur quadratique moyenne :

$$\mathcal{L}_{\text{rec}}\Big(\mathbf{x}_{a,s}; \boldsymbol{\phi}, \boldsymbol{\theta}\Big) = \sum_{i=1}^{d} (x_{a,s}^{(i)} - \hat{x}_{a,s}^{(i)})^2. \tag{3.41}$$

Dans le cadre de notre méthode de sous-échantillonnage, puisque nous devons obtenir une représentation latente $\mathbf{z}_{a,s}\ \forall\ \mathbf{x}_{a,s} \in \mathcal{D}^{-}_{fiable}$ et que $\mathcal{D}^{-}_{fiable}$ est le seul ensemble de données à notre disposition pour l'entraînement, nous employons deux auto-encodeurs ayant les paramètres optimaux suivants :

$$\text{Auto-encodeur 1} : (\boldsymbol{\phi}_1^*, \boldsymbol{\theta}_1^*) = \arg\min_{\boldsymbol{\phi}_1, \boldsymbol{\theta}_1} \frac{1}{|\mathcal{D}^{-}_{1;fiable}|} \sum_{(a,s) \in \mathcal{D}^{-}_{1;fiable}} \mathcal{L}_{\text{rec}}\Big(\mathbf{x}_{a,s}; \boldsymbol{\phi}_1, \boldsymbol{\theta}_1\Big), \tag{3.42}$$

$$\text{Auto-encodeur 2} : (\boldsymbol{\phi}_2^*, \boldsymbol{\theta}_2^*) = \arg\min_{\boldsymbol{\phi}_2, \boldsymbol{\theta}_2} \frac{1}{|\mathcal{D}^{-}_{2;fiable}|} \sum_{(a,s) \in \mathcal{D}^{-}_{2;fiable}} \mathcal{L}_{\text{rec}}\Big(\mathbf{x}_{a,s}; \boldsymbol{\phi}_2, \boldsymbol{\theta}_2\Big). \tag{3.43}$$

Dans ces expressions, les ensembles $\mathcal{D}^{-}_{1;fiable}$ et $\mathcal{D}^{-}_{2;fiable}$ sont mutuellement exclusifs et contiennent

chacun une moitié des observations de $\mathcal{D}^{-}_{fiable}$. L'auto-encodeur 1 est ensuite utilisé pour obtenir les représentations latentes $\mathbf{z}_{a,s}$ correspondantes aux observations $\mathbf{x}_{a,s} \in \mathcal{D}^{-}_{2;fiable}$ et vice-versa. La figure 3.10 offre une visualisation de la structure de l'auto-encodeur employé.

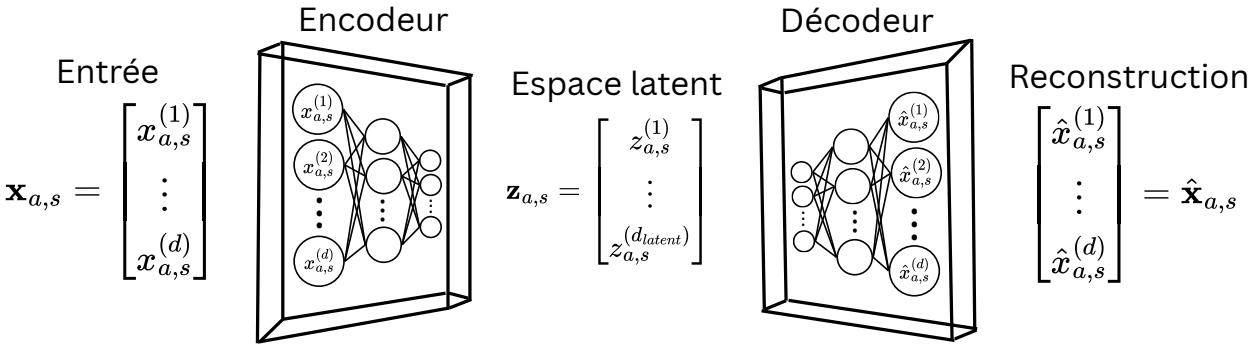

FIGURE 3.10 Illustration de l'auto-encodeur employé pour réduire la dimensionnalité la représentation des acides aminés.

Les auto-encodeurs sont des modèles puissants qui permettent d'apprendre des relations non linéaires entre les variables, mais peuvent être vulnérables au surapprentissage s'ils ne sont pas régularisés correctement. Dans notre cas, s'il y a surapprentissage, les caractéristiques latentes apprises risquent de ne pas être pertinentes, ce qui peut nuire à l'étape subséquente de regroupement par k-moyennes, et ultimement, réduire la qualité des échantillons d'observations négatives sélectionnés pour entraîner notre modèle. Afin d'éviter le surapprentissage, nous ajoutons un terme de régularisation de Ridge à la fonction de perte comme suit :

$$\mathcal{L}_{\text{rec}}\Big(\mathbf{x}_{a,s}; \boldsymbol{\phi}, \boldsymbol{\theta}\Big) = \sum_{i=1}^{d}(x_{a,s}^{(i)} - \hat{x}_{a,s}^{(i)})^2 + \lambda \sum_{i=1}^{d} \beta_i^2. \tag{3.44}$$

Dans le terme ajouté, β_i représente les paramètres du modèle et λ est un hyperparamètre qui contrôle la force de la régularisation. En plus de l'ajout de ce terme de régularisation, nous employons également la technique de « dropout », qui consiste à ignorer certains neurones d'une couche de manière aléatoire pendant l'entraînement, et ce, au sein de l'encodeur et du décodeur. En effet, les neurones d'un réseau neuronal peuvent s'adapter conjointement, ce qui signifie qu'ils peuvent développer des dépendances les uns par rapport aux autres. La technique « dropout » réduit le risque de développer ce type de dépendance en veillant à ce que chaque neurone fonctionne de manière plus indépendante.

3.3.5 Modèles prédictifs

Dans cette section, nous présentons les modèles de classification que nous employons pour prédire la phosphorylation des acides aminés. Les modèles abordés sont introduits dans un ordre soigneusement choisi afin de mettre en lumière le cheminement logique qui les relie : du plus simple au plus complexe, tout en soulignant les liens conceptuels qui existent entre eux.

Analyse par discriminant linéaire

L'analyse discriminante linéaire (LDA) est une méthode de classification statistique, de type génératif, qui recherche une combinaison linéaire de caractéristiques pour séparer les classes. Dans notre cas, la LDA recherche l'hyperplan qui établit la meilleure discrimination entre les acides aminés phosphorylés et ceux qui ne le sont pas.

Pour illustrer comment un modèle LDA s'insère dans notre problématique, considérons un exemple fictif dans lequel un échantillon est constitué de 100 résidus de sérine, parmi lesquels 50 résidus sont phosphorylés et 50 ne le sont pas. L'objectif est de prédire l'état de phosphorylation à partir d'une caractéristique hypothétique, soit la polarité moyenne des résidus voisins, mesurée sur une échelle allant de 1 à 10. Il est établi, d'après des connaissances empiriques, que les sites phosphorylés se situent généralement dans des environnements plus polaires, correspondant à un indice moyen de polarité égal à 8, tandis que l'environnement des sites non phosphorylés présente un indice moyen de polarité de 2. Ainsi, si la polarité moyenne mesurée de l'environnement du résidu d'intérêt est égale à 9, la probabilité que ce résidu soit phosphorylé apparaît élevée. À l'inverse, une polarité mesurée à 5, valeur intermédiaire entre les deux indices moyens mentionnés, rendrait la décision moins évidente. Cet exemple fictif illustre le principe fondamental de l'analyse discriminante linéaire, lorsqu'il est adapté à notre problématique.

Formellement, un modèle LDA suppose que pour chaque classe k, les caractéristiques sont distribuées selon une distribution normale multivariée avec un vecteur moyen propre à chaque classe et une matrice de covariance commune à toutes les classes. Un classificateur LDA utilise la proportion de chaque classe k dans les données d'apprentissage pour estimer des probabilités a priori : $\pi_k = P[Y = k]$. On remarque directement l'importance de balancer le nombre d'acides aminés phosphorylés et non phosphorylés pour l'utilisation du modèle LDA.

Soit les fonctions de densité conditionnelles de chaque classe $f_k(x)$ et les probabilités a priori π_k, LDA applique la règle de Bayes pour calculer la probabilité a posteriori qu'un acide aminé

appartienne à la classe k comme suit :

$$P[Y = k \mid X = x] = \frac{f_k(x)\pi_k}{P[X = x]}. \tag{3.45}$$

Dans l'expression ci-haut, puisque le terme $P[X = x]$ ne dépend pas de la classe k, on peut étudier les probabilités a posteriori en se concentrant sur la vraisemblance de chaque classe : $f_k(x)\pi_k$, et cheminer jusqu'aux fonctions objectives $\delta_k(x)$:

$$\delta_k(x) = \log \pi_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + x^T\Sigma^{-1}\mu_k. \tag{3.46}$$

Pour faire la prédiction, on choisit la classe k qui maximise $\delta_k(x)$:

$$\hat{y} = \underset{k}{\operatorname{argmax}}\ \delta_k(x). \tag{3.47}$$

Le modèle LDA est simple et offre une bonne interprétabilité, grâce à l'utilisation des probabilités a priori. Ce genre de modèle est moins exposé au risque de surapprentissage, dû à sa faible variance. Cependant, l'approche fait l'hypothèse que les données de chaque classe sont distribuées normalement dans l'espace des caractéristiques et que toutes les classes partagent la même matrice de covariance. Les modèles LDA assument également que chaque caractéristique contribue de façon linéaire à la frontière de décision, donc si la vraie frontière de décision est hautement non linéaire, LDA peut s'avérer un mauvais choix. Dans notre contexte, nous ne croyons pas que le modèle LDA produise les meilleurs résultats. En effet, il est très fréquent dans les contextes biologiques/moléculaires que des phénomènes hautement non linéaires soient observés et que les données ne soient pas normalement distribuées. La LDA est cependant un modèle très rapide à entraîner et facile à interpréter, en faisant un bon modèle de base pour itérer et tester rapidement lors de l'exploration de différentes stratégies préliminaires comme le prétraitement de données, l'échantillonnage et le balancement des classes ou tester l'intégration de nouvelles caractéristiques au modèle.

Régression logistique

La régression logistique adopte une approche discriminative pour faire la classification. Contrairement au modèle LDA, les probabilités $P[Y \mid X]$ sont donc estimées sans faire d'hypothèse sur la distribution des caractéristiques employées pour modéliser les acides aminés. En effet, la régression logistique modélise directement la probabilité d'observer la classe positive,

conditionnellement aux caractéristiques $\mathbf{x}_i$, en utilisant la fonction sigmoïde :

$$P[y_i = 1 \mid \mathbf{x}_i; \boldsymbol{\beta}] = \frac{1}{1 + \exp(-\mathbf{x}_i \boldsymbol{\beta})}. \tag{3.48}$$

Pour notre projet, nous utilisons une fonction de perte d'entropie croisée binaire pondérée en fonction de la classe du résidu :

$$\min_{\boldsymbol{\beta}} \frac{1}{S} \sum_{i=1}^{n} s_i \Big[-y_i \log\Big(\hat{p}(x_i)\Big) - (1 - y_i) \log\Big(1 - \hat{p}(x_i)\Big) \Big] + \lambda \frac{r(\boldsymbol{\beta})}{S}, \tag{3.49}$$

où s_i est le poids attribué au résidu, en fonction de sa classe et $S = \sum_{i=1}^{n} s_i$. Les poids s_i peuvent prendre 2 valeurs (hyperparamètres), en fonction de la classe de l'acide aminé. L'ajout de ces poids permet de moduler l'apprentissage en pénalisant davantage les erreurs sur les acides aminés phosphorylés.

λ est un hyperparamètre contrôlant la force du terme de régularisation $r(\boldsymbol{\beta})$. Dans notre contexte, il est pertinent de se pencher sur la différence entre les modèles discriminatifs et génératifs, pour identifier les risques liés au surapprentissage. Pour illustrer cette vulnérabilité et pourquoi elle est importante à considérer dans notre cas, considérons un scénario dans lequel on fournit en entrée au classificateur le vecteur $\mathbf{x}_{a,s}$ d'un résidu, en plus d'une image du résidu en question, dont les couleurs des pixels sont extraites. Dans cette situation, un modèle génératif aurait pour objectif de modéliser les distributions des différentes caractéristiques pour les deux classes. Ainsi, à partir d'un résidu de test, le modèle génératif se demande si c'est la distribution correspondant aux résidus phosphorylés ou non phosphorylés qui correspond le mieux au résidu qui lui est présenté, et il choisit cette étiquette. En revanche, un modèle discriminatif tente plutôt d'identifier la ou les combinaisons de caractéristiques permettant de maximiser les performances. À titre d'exemple, si tous les résidus non phosphorylés dans les données d'apprentissage sont jaunes et qu'aucun résidu phosphorylé ne l'est, cette seule caractéristique suffirait à séparer les classes, et le modèle s'en satisferait, le rendant vulnérable au surapprentissage. Les types de régularisation qui seront employés pour éviter le surapprentissage sont présentés dans le tableau ci-dessous.

À partir du tableau, on remarque qu'en prenant les valeurs absolues des coefficients β, la régularisation de Lasso (ℓ_1) réduit les poids des variables moins utiles à 0, ce qui explique pourquoi nous utilisons aussi la régularisation de Lasso pour la sélection de variables. D'autre part, la régularisation de Ridge (ℓ_2) a comme effet de pénaliser les valeurs des coefficients élevées, mais sans que ceux-ci atteignent la valeur nulle. Finalement, ElasticNet est une technique hybride équilibrant les forces de ℓ_1 et ℓ_2 via l'hyperparamètre ρ.

La minimisation de la fonction objectif s'effectue à l'aide de la descente de gradient, où les

TABLEAU 3.2 Formulations des différents termes de régularisation $r(\boldsymbol{\beta})$

Type de régularisation	$r(\boldsymbol{\beta})$
Aucune	0
ℓ_1	$\|\boldsymbol{\beta}\|_1$
ℓ_2	$\frac{1}{2}\|\boldsymbol{\beta}\|_2^2$
ElasticNet	$\frac{1-\rho}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta} + \rho\|\boldsymbol{\beta}\|_1$

paramètres sont mis à jour itérativement dans la direction opposée au gradient de la fonction de perte. Le gradient de $\mathcal{L}(\boldsymbol{\beta})$ par rapport à $\boldsymbol{\beta}$, noté $\nabla_\beta \mathcal{L}(\boldsymbol{\beta})$, s'exprime comme suit :

$$\nabla_\beta \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{S}\sum_{i=1}^{n} s_i\Big(\hat{p}(\mathbf{x}_i) - y_i\Big)\mathbf{x}_i^T + \lambda\frac{\nabla_\beta r(\boldsymbol{\beta})}{S}, \tag{3.50}$$

où $\hat{p}(\mathbf{x}_i)$ est la probabilité prédite pour l'observation i, $(\hat{p}(\mathbf{x}_i) - y_i)$ représente l'erreur de prédiction pour l'acide aminé, et $\nabla_\beta r(\boldsymbol{\beta})$ est le gradient. Nous utilisons la régression logistique dans notre projet car elle offre une bonne balance entre complexité et simplicité. En effet, elle permet de modéliser des relations plus complexes entre les variables que LDA, tout en gardant un niveau de variance plus bas que le prochain type de modèle, les réseaux de neurones.

Réseaux de neurones artificiels

Les réseaux de neurones artificiels (RNA) entretiennent un lien étroit avec la régression logistique, mais leur architecture plus sophistiquée leur confère la capacité d'apprendre des relations plus complexes entre les variables, ce qui est désirable dans un contexte biologique. Pour comprendre de manière intuitive pourquoi, imaginons la régression logistique comme un modèle simplifié d'un neurone biologique. En utilisant cette analogie, les RNA pourraient être vus comme un cerveau, soit un ensemble de neurones connectés les uns aux autres. La figure ci-dessous offre une visualisation de cette analogie. Ainsi, les RNA peuvent être interprétés comme le résultat « d'empiler » plusieurs couches de fonctions de régression logistique. La combinaison des multiples unités individuelles et le concept de profondeur (couches cachées) est ce qui donne aux réseaux de neurones la capacité de modéliser des relations non linéaires et plus complexes que la régression logistique.

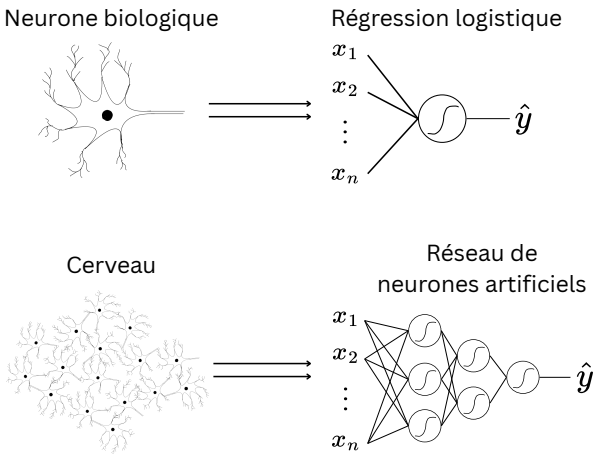

FIGURE 3.11 Figure mettant en relation la régression logistique et les réseaux de neurones via une analogie avec le cerveau humain.

Voici l'architecture du modèle RNA employé dans notre étude :

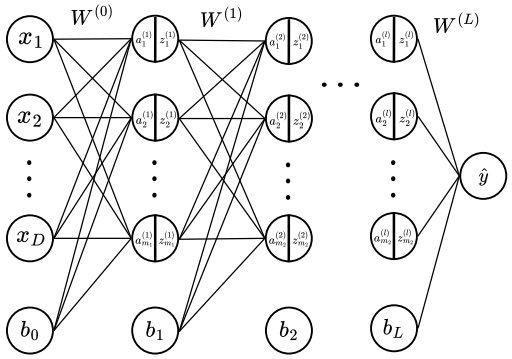

FIGURE 3.12 Architecture du modèle RNA employé dans notre étude.

Les termes employés dans la figure sont définis comme suit :

- $\mathbf{x} = [x_1, x_2, ..., x_D]^T$ est le vecteur d'entrée de dimension $D = 223$.
- L est le nombre total de couches dans le réseau.
- m_l est le nombre de neurones dans la couche l.
- $\mathbf{W}^{(l)}$ est la matrice des poids pour la couche l. Sa dimension est $m_l \times m_{l-1}$.
- $\mathbf{b}^{(l)} \in \mathbb{R}^{m_l}$ est le vecteur de biais pour la couche l.
- $\mathbf{a}^{(l)} \in \mathbb{R}^{m_l}$ est le vecteur des pré-activations pour la couche l.
- $\mathbf{z}^{(l)} \in \mathbb{R}^{m_l}$ est le vecteur des activations pour la couche l.
- $h^{(l)}(\cdot)$ est la fonction d'activation appliquée à la couche l.

Le processus de passe avant peut être décrit itérativement pour chaque couche l comme suit :

$$\mathbf{a}^{(l)} = \mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)} \tag{3.51}$$

$$\mathbf{z}^{(l)} = h^{(l)}(\mathbf{a}^{(l)}) \tag{3.52}$$

Ce processus se répète pour toutes les couches jusqu'à la couche finale L. La sortie finale du réseau est le vecteur d'activation de la dernière couche, $\hat{\mathbf{y}} = \mathbf{z}^{(L)}$. Dans notre cas, la couche de sortie ne contient qu'un seul neurone. La fonction d'activation $h^{(L)}$ est la fonction sigmoïde, car elle produit une sortie dans l'intervalle $[0, 1]$, représentant la probabilité de phosphorylation. Le choix des fonctions d'activation $h^{(l)}$ pour les couches cachées est un hyperparamètre du modèle. Voici un tableau des fonctions d'activation qui seront testées dans notre projet :

Tableau 3.3 Fonctions d'activation employées pour le modèle RNA.

Fonction d'activation	**Formule**	**Commentaire**
Sigmoïde	$\sigma(x) = \dfrac{1}{1+e^{-x}}$	Pour couche de sortie, disparition du gradient dans couches cachées.
Tanh	$\tanh(x) = \dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	Retourne une valeur dans $[-1, 1]$. Centrée autour de 0.
ReLU	$\mathrm{ReLU}(x) = \max(0,\, x)$	Non linéaire et simple, problème des neurones morts si $x \leq 0$.
Leaky ReLU	$\mathrm{LeakyReLU}(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases}$	Évite les neurones morts (introduit pente α si $x \leq 0$)

Pour définir les valeurs optimales des matrices de poids $\mathbf{W}^{(l)}$ et des vecteurs de biais $\mathbf{b}^{(l)}$, on minimise la fonction de perte suivante :

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]. \tag{3.53}$$

Pour trouver les poids et biais qui minimisent $\mathcal{L}$, la descente de gradient stochastique (SGD) est employée. Cet algorithme se sert de la technique de rétropropagation pour le calcul du gradient de la fonction de perte par rapport à chaque paramètre du réseau. On obtient ainsi

les gradients :

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)} (\mathbf{z}^{(l-1)})^T \tag{3.54}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)} \tag{3.55}$$

Les RNA sont employés dans notre étude pour leur capacité à modéliser des relations complexes entre les variables, ce qui est pertinent dans un contexte biologique comme le nôtre. Toutefois, sans régularisation appropriée, ces modèles sont susceptibles de souffrir de surapprentissage. C'est pourquoi nous utilisons également les 3 méthodes de régularisation présentées dans la section portant sur la régression logistique ($\ell 1, \ell 2$, et Elastic Net), dans la fonction de perte des RNA. En surplus, nous employons aussi la technique de régularisation « dropout », déjà abordée dans la section des auto-encodeurs.

Forêts aléatoires

Les forêts aléatoires sont des modèles de la famille des méthodes d'ensemble, où un modèle est le résultat de l'agrégation de plusieurs modèles simples. Plus spécifiquement, une forêt aléatoire comprend un ensemble de T arbres de décision. Un arbre de décision est un organigramme formé d'une séquence de règles de décision permettant de séparer progressivement les observations selon leurs caractéristiques. Un arbre est construit à partir d'un échantillon d'observations tiré avec remise depuis l'ensemble d'entraînement. Ce sous-échantillonnage directement intégré à la phase d'apprentissage est ce qui confère aux forêts aléatoires une variance plus faible, les rendant plus robustes au surapprentissage, une caractéristique désirable dans un contexte biologique. Lors de la construction d'un arbre, on cherche à partitionner l'espace des caractéristiques suivant un critère d'impureté, en sélectionnant de manière aléatoire un sous-ensemble de variables et en établissant des points de séparation. Dans notre contexte, cela correspond à séparer les échantillons en fonction d'un sous-ensemble de caractéristiques du vecteur $\mathbf{x}_{a,s}$.

Pour établir les points de séparation dans les nœuds, les deux métriques employées sont le coefficient de Gini et l'entropie, définis respectivement par :

$$\text{Gini} = 1 - \sum_{i=1}^{k} p_i^2, \tag{3.56}$$

$$\text{Entropie} = -\sum_{i=1}^{k} p_i \log_2(p_i). \tag{3.57}$$

p_i est la proportion d'acides aminés appartenant à la classe i dans le nœud considéré, et k est

le nombre de classes. Pour déterminer le meilleur point de séparation d'un nœud, on cherche à maximiser la réduction de l'impureté. En utilisant le coefficient de Gini ou l'entropie comme mesure d'impureté $I(\cdot)$, le gain d'une séparation qui divise un ensemble d'acides aminés S (nœud parent) en m sous-ensembles $S_1, S_2, ..., S_m$ (nœuds enfants) est :

$$\text{Gain}(S, \text{Séparation}) = I(S) - \sum_{v=1}^{m} \frac{|S_v|}{|S|} I(S_v). \tag{3.58}$$

$I(S)$ est l'impureté de l'ensemble S avant la séparation. S_v est le sous-ensemble d'échantillons allant vers l'enfant v après la séparation.

Au-delà de son rôle dans la construction des arbres, le gain d'impureté associé à chaque variable peut être agrégé sur l'ensemble de la forêt pour estimer l'importance globale de cette variable pour la prédiction. On peut donc obtenir une mesure de la capacité d'une caractéristique à discriminer les acides aminés phosphorylés et ceux qui ne le sont pas. Cela est utile dans notre contexte car nous avons plusieurs propriétés physico-chimiques agrégées selon différents rayons autour de l'acide aminé d'intérêt. Ces mesures de réduction d'impureté permettent ainsi d'établir un seuil (rayon) à partir duquel il devient superflu de prendre en compte les valeurs des propriétés physico-chimiques des acides aminés voisins. Nous utiliserons d'ailleurs cette approche pour la sélection de variables.

Afin d'estimer la probabilité qu'un acide aminé représenté par $\mathbf{x}_{a,s}$ soit phosphorylé, on agrège les sorties de chaque arbre. Plus précisément, si l'arbre t produit une évaluation $\hat{p}_t(\mathbf{x}_{a,s})$, la probabilité de phosphorylation est obtenue via une moyenne :

$$\hat{p}[\mathbf{x}_{a,s}] = \frac{1}{T} \sum_{t=1}^{T} \hat{p}_t[\mathbf{x}_{a,s}]. \tag{3.59}$$

Ce procédé offre une estimation plus stable que celle d'un arbre unique, car la variance de l'estimateur global tend à diminuer lorsque les arbres individuels ne sont pas entièrement corrélés. Une forêt aléatoire est particulièrement adaptée lorsque le nombre de variables est grand et que les interactions sont susceptibles d'être hautement non linéaires, ce qui est notre cas, car plusieurs variables (position dans la séquence, propriétés physico-chimiques, variables de densité) peuvent interagir de manière subtile.

Un arbre de décision laissé libre de croître jusqu'à ce que chaque feuille soit pure peut devenir trop profond et complexe, menant au surapprentissage. Pour éviter cela, il existe deux stratégies. La première consiste à arrêter prématurément la croissance de l'arbre en définissant des critères d'arrêt, tels qu'une profondeur maximale, un nombre minimal d'échantillons requis pour diviser un nœud, ou un seuil minimal d'amélioration de l'impureté. La seconde straté-

gie, l'élagage (pruning), implique de construire d'abord un arbre complet, puis de supprimer récursivement les branches ayant peu de pouvoir discriminatif sur un ensemble de validation ou via une pénalisation de la complexité. Même si l'élagage améliore les performances, son coût de calcul, très élevé dès qu'on manipule un grand nombre d'observations à haute dimensionnalité, comme c'est notre cas, constitue une barrière à son utilisation dans notre projet.

XGBoost

Après les forêts aléatoires, nous abordons une autre approche d'ensemble basée sur les arbres : le boosting de gradient, et plus spécifiquement son implémentation optimisée XGBoost (Extreme Gradient Boosting). Contrairement aux forêts aléatoires, le boosting est une méthode séquentielle où chaque nouvel arbre est entraîné pour corriger les erreurs résiduelles des arbres précédents. Le modèle est construit itérativement comme suit :

$$\hat{y}_i^{(0)} = 0 \tag{3.60}$$
$$\hat{y}_i^{(1)} = \hat{y}_i^{(0)} + f_1(\mathbf{x}_i) \tag{3.61}$$
$$\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + f_2(\mathbf{x}_i) = f_1(\mathbf{x}_i) + f_2(\mathbf{x}_i) \tag{3.62}$$
$$\vdots \tag{3.63}$$
$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i) = \sum_{k=1}^{t} f_k(\mathbf{x}_i) \tag{3.64}$$

L'objectif à chaque étape t est de trouver la fonction f_t qui minimise la fonction de perte globale. Faisant encore usage de plusieurs modèles faibles, XGBoost est une méthode moins susceptible de souffrir de surapprentissage et plus résistante au bruit. Cette robustesse au bruit est avantageuse dans notre contexte, car parmi les 24 caractéristiques obtenues de l'agrégation d'une même propriété physico-chimique, il est probable que certaines incluent de l'information provenant d'acides aminés situés trop loin pour influencer réellement la probabilité de phosphorylation de l'acide aminé étudié.

Un facteur expliquant la robustesse du modèle XGboost face au surapprentissage est le fait que sa fonction objectif globale $\mathcal{O}$ combine la fonction perte, notée L, et un terme de régularisation Ω pénalisant la complexité des modèles :

$$\mathcal{O} = \sum_{i=1}^{N} L(y_i, \hat{y}_i^{(T)}) + \sum_{t=1}^{T} \Omega(f_t). \tag{3.65}$$

T est le nombre total d'arbres, y_i la vraie étiquette, et $\hat{y}_i^{(T)}$ la prédiction. Pour déterminer

la fonction f_t à ajouter à l'étape t, XGBoost procède à la minimisation de l'objectif à cette étape spécifique. En utilisant $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)$, l'objectif à l'étape t peut s'écrire ainsi :

$$\mathcal{O}^{(t)} \approx \sum_{i=1}^{N} \left[L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \gamma T_t + \frac{1}{2} \lambda \sum_{j=1}^{T_t} w_j^2. \tag{3.66}$$

Ici, g_i et h_i sont respectivement le gradient et le hessien de la fonction de perte L par rapport à $\hat{y}_i^{(t-1)}$, évalués en $\hat{y}_i^{(t-1)}$. T_t est le nombre de feuilles dans l'arbre t, w_j est le poids associé à la j-ème feuille, et γ et λ sont des hyperparamètres de régularisation. La valeur $\mathcal{O}^{(t)}$ sert de score pour mesurer la qualité des arbres. Plus spécifiquement, une valeur plus faible indique une meilleure structure, car elle représente un arbre qui corrige efficacement les erreurs résiduelles précédentes tout en restant suffisamment simple pour généraliser et éviter le surapprentissage. Dans notre contexte, ce score sera plus faible pour les arbres qui identifient des règles pertinentes pour la prédiction de la phosphorylation, sans pour autant surapprendre les spécificités de l'ensemble d'entraînement. Une fois l'ensemble des T arbres appris, la prédiction pour une observation est donnée par :

$$\hat{y}_i = \sum_{t=1}^{T} f_t(\mathbf{x}_i). \tag{3.67}$$

3.3.6 Métriques de performance

Nous présenterons dans cette section les 6 métriques que nous jugeons pertinentes, afin de pouvoir évaluer correctement les performances de prédiction de la phosphorylation des acides aminés. Ces métriques s'appuient sur les valeurs suivantes : un vrai positif (TP) correspond à un acide aminé réellement phosphorylé que le modèle prédit correctement comme phosphorylé, et un vrai négatif (TN) est un acide aminé réellement non phosphorylé que le modèle prédit correctement comme non phosphorylé. Inversement, un faux positif (FP) se produit lorsqu'un acide aminé non phosphorylé est incorrectement prédit comme phosphorylé, et un faux négatif (FN) lorsqu'un acide aminé réellement phosphorylé est prédit comme non phosphorylé.

1. Exactitude

L'exactitude représente la proportion de prédictions correctes (positives et négatives) parmi toutes les classifications. Elle est définie comme suit :

$$\text{Exactitude} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.68}$$

Bien qu'elle donne une idée générale des performances d'un modèle, son utilité diminue lorsque les classes sont déséquilibrées. Dans notre cas, un modèle pourrait atteindre une exactitude élevée en prédisant la classe phosphorylée la plupart du temps. Néanmoins, puisque nous procédons à un balancement des classes, elle reste une métrique utile.

2. Précision

Dans notre contexte, la précision indique le nombre de résidus prédits comme phosphorylés et qui le sont réellement. Elle se calcule comme suit :

$$\text{Précision} = \frac{TP}{TP + FP}. \tag{3.69}$$

Une précision élevée signifie que lorsque le modèle signale un site comme étant phosphorylé, il est souvent correct. Cette mesure est importante si les faux positifs sont coûteux ou longs à étudier, ce qui est le cas de la quantification expérimentale de la phosphorylation en laboratoire.

3. Rappel

Le rappel mesure la proportion de résultats positifs réels que le modèle identifie correctement. Il nous informe donc sur la proportion de résidus réellement phosphorylés qui ont été correctement classifiés. On écrit le rappel de la façon suivante :

$$\text{Rappel} = \frac{TP}{TP + FN}. \tag{3.70}$$

Dans notre cas, le suivi de cette métrique est crucial, car un faible rappel indique que le modèle échoue à identifier certains mécanismes de phosphorylation clés. Un tel modèle aura alors de la difficulté à repérer les acides aminés susceptibles d'être phosphorylés qui n'ont jamais été étudiés expérimentalement, limitant ainsi sa capacité à découvrir de nouveaux biomarqueurs potentiels.

4. Score F-1

Le score F1 est la moyenne harmonique de la précision et du rappel :

$$\text{Score-F1} = 2 \times \frac{\mathit{Précision} \times \mathit{Rappel}}{\mathit{Précision} + \mathit{Rappel}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.71}$$

Parce qu'il équilibre la précision et le rappel, le score F1 est bien adapté aux scénarios déséquilibrés comme le nôtre. La moyenne harmonique pénalise davantage les valeurs extrêmes, fournissant ainsi une métrique qui garantit que ni la précision ni le rappel ne dominent au détriment de l'autre.

5. ROC-AUC

La ROC-AUC quantifie l'aire sous la courbe formée par le tracé du taux de vrais positifs par rapport au taux de faux positifs au fur et à mesure que le seuil de classification varie. On peut calculer sa valeur de la manière suivante :

$$\text{ROC-AUC} = \int_0^1 \text{TPR}\ d(\text{FPR}) \tag{3.72}$$

La courbe ROC montre la capacité du modèle à distinguer les sites phosphorylés des sites non phosphorylés en fonction de différents seuils de décision. Bien que la courbe ROC-AUC soit largement utilisée et fournisse une vue d'ensemble des performances du modèle, elle peut parfois être trop optimiste dans les ensembles de données déséquilibrés si le nombre de vrais négatifs est élevé. Cependant, l'étape de balancement des classes devrait contribuer à atténuer ce problème.

6. PR-AUC

L'aire sous la courbe précision-rappel (PR-AUC) est bien adaptée pour évaluer la performance sur la classe positive dans un contexte de déséquilibre des classes. Cette métrique se concentre sur le maintien d'une haute précision tout en maximisant le rappel, ce qui correspond au besoin d'identifier des résidus susceptibles d'être phosphorylés, tout en limitant les faux positifs. Puisque notre modèle sert à identifier des sites de phosphorylation pertinents à étudier en laboratoire, un faux positif entraîne un travail expérimental inutile. Privilégier la PR-AUC permet donc de mettre l'accent sur la fiabilité des prédictions positives. On écrit cette métrique comme suit :

$$\text{PR-AUC} = \int_0^1 \text{Précision}\ d(\text{Rappel}). \tag{3.73}$$

CHAPITRE 4 RÉSULTATS

La première section de ce chapitre porte sur la présentation des résultats obtenus à chaque étape de l'approche systématique d'apprentissage machine, ainsi qu'une comparaison entre notre modèle et ceux issus de la littérature. La deuxième section est consacrée à l'analyse des résultats obtenus.

4.1 Présentation des résultats

Dans cette section, nous présentons séquentiellement les résultats de classification obtenus à chacune des étapes de notre approche systématique d'apprentissage machine. Les résultats correspondent aux moyennes des différentes métriques de performance, calculées à partir d'une validation croisée imbriquée répétée 10 fois, générant ainsi 50 résultats de tests. Étant donné que la PR-AUC a été identifiée comme une métrique clé pour notre étude, seuls les graphiques et tableaux relatifs à cette métrique sont inclus dans cette section. Les résultats concernant les autres métriques, ainsi que les détails relatifs à la configuration des hyperparamètres des modèles utilisés, figurent en annexe.

4.1.1 Normalisation des données

Dans un premier temps, nous avons testé les différentes méthodes de normalisation. Les classes ont été balancées de manière aléatoire et aucune méthode de sélection de variables n'a été utilisée. La figure ci-bas expose les résultats.

On observe que la performance augmente dès qu'on applique une normalisation quelconque pour le modèle RNA et la régression logistique. D'autre part, on observe moins d'impact pour les modèles de forêt aléatoire, de LDA et de XGBoost. Ce comportement est respectif à nos attentes, car la régression logistique et les réseaux de neurones utilisent des procédures d'optimisation basées sur le gradient, qui sont plus sensibles à l'échelle et à la distribution des variables. À l'inverse, les méthodes basées sur des arbres (forêt aléatoire, XGBoost) et l'approche LDA sont reconnues pour être moins sensibles à l'échelle des données.

Il est difficile de trancher entre les différentes méthodes de normalisation, car il ne semble pas y avoir de différence significative entre les méthodes. Nous choisissons la standardisation pour la suite des résultats, car cette méthode est particulièrement bien adaptée pour les réseaux de neurones et ceux-ci sont utilisés dans l'auto-encodeur employé dans notre méthode originale d'échantillonnage et de balancement des classes. Voici maintenant les résultats des phases

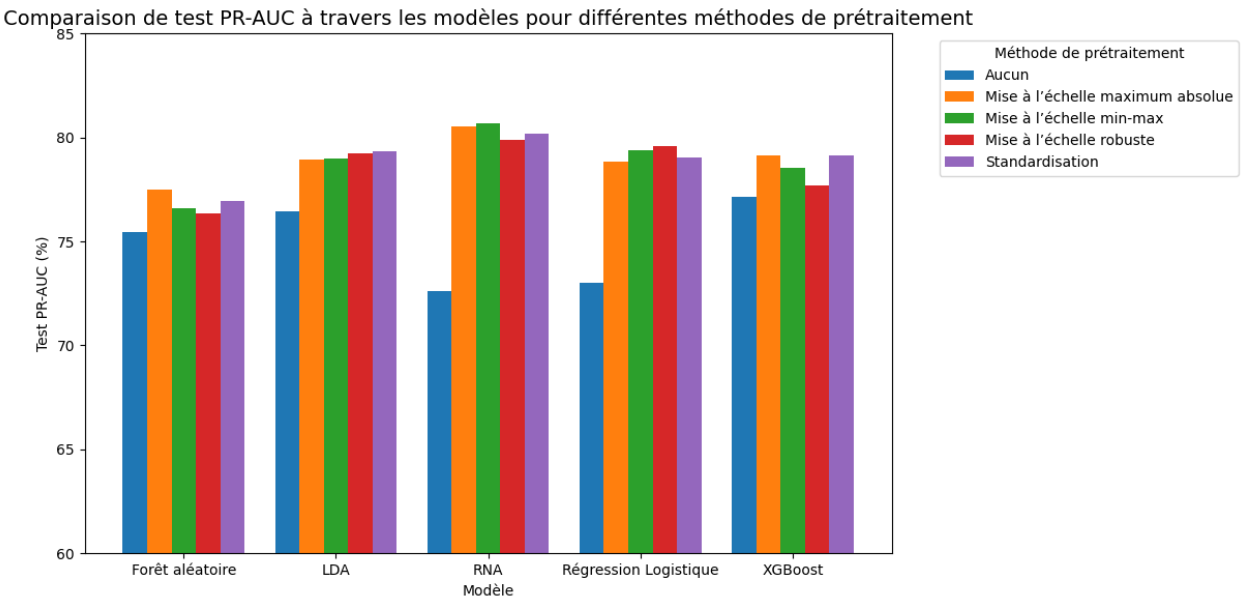

FIGURE 4.1 AUC-PR des différents modèles avec différentes normalisations.

d'entraînement, de test et de validation, avec standardisation. Des méthodes de régularisation

TABLEAU 4.1 Comparaison des modèles selon l'AUC-PR en entraînement, test et validation croisée (VC), après standardisation.

Modèle	**Entraînement AUC-PR**	**VC AUC-PR**	**Test AUC-PR**
Régression Logistique	0.8032 (0.0065)	0.7993 (0.0084)	0.7892 (0.0081)
RNA	**0.8109 (0.0064)**	**0.8020** (0.0078)	**0.8093** (0.0089)
XGBoost	0.8004 (0.0067)	0.7935 (0.0084)	0.7910 (0.0082)
Forêt aléatoire	0.7836 (0.0069)	0.7654 (0.0082)	0.7698 (0.0083)
LDA	0.8056 (0.0062)	0.7979 (0.0079)	0.7935 (0.0077)

ont été employées pour les modèles RNA, forêt aléatoire, XGBoost et la régression logistique. Le fait que les performances d'entraînement soient similaires aux performances de test suggère que les modèles ne souffrent pas de surapprentissage. On note aussi que la variance des résultats est plutôt faible.

4.1.2 Échantillonnage et balancement des classes

Afin d'évaluer notre méthode originale d'échantillonnage et de balancement des classes, nous comparons ses performances avec celles de 3 approches utilisées dans la littérature : CD-HIT, la méthode bootstrap et un balancement aléatoire. Pour obtenir les résultats, les données ont

été mises à l'échelle avec une standardisation et aucune méthode de sélection de variables n'a été employée. Les résultats montrent que la méthode originale améliore la capacité de

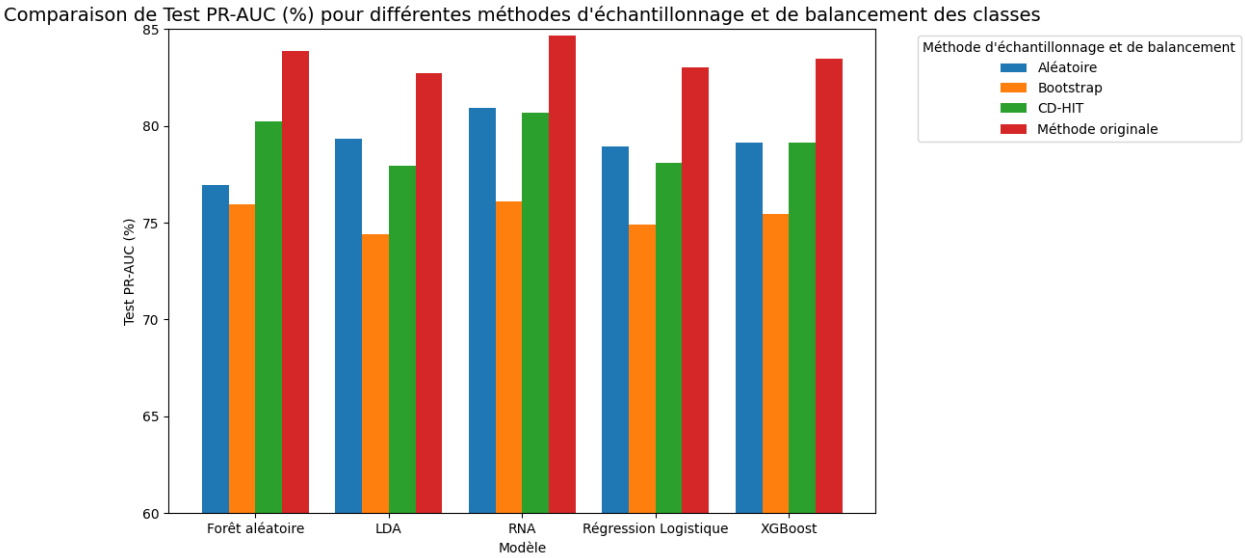

FIGURE 4.2 Comparaison de l'AUC-PR pour différentes méthodes d'échantillonnage et de balancement des classes en entraînement, test et validation croisée (VC)

généralisation de tous les modèles. En particulier, les modèles capables de saisir des relations non linéaires complexes (RNA, forêt aléatoire et XGBoost) semblent en profiter davantage. Ceci peut s'expliquer par le fait que notre méthode permet d'obtenir un échantillon plus représentatif et diversifié de la classe négative, aidant ainsi ces modèles à exploiter leur plein potentiel. On remarque aussi que notre méthode permet de réduire de manière importante la variance des résultats. Cela s'explique par le fait que nous intégrons explicitement une étape de réduction du bruit dans notre méthode.

Tableau 4.2 Comparaison de l'AUC-PR pour différentes méthodes d'échantillonnage et de balancement des classes en entraînement, test et validation croisée (VC)

Échantillonnage et balancement des classes	**Modèle**	**Entraînement PR-AUC**	**VC PR-AUC**	**Test PR-AUC**
Méthode originale	Régression Logistique	0.8407 (0.0021)	0.8284 (0.0031)	0.8352 (0.0039)
	RNA	0.8502 (0.0022)	0.8571 (0.0033)	**0.8568 (0.0034)**
	XGBoost	**0.8775 (0.0023)**	**0.8599 (0.0031)**	0.8317 (0.0038)
	Forêt aléatoire	0.8559 (0.0045)	0.8380 (0.0034)	0.8408 (0.0039)
	LDA	0.8221 (0.0040)	0.8191 (0.0035)	0.8071 (0.0049)
Bootstrap	Régression Logistique	0.7809 (0.0292)	0.7793 (0.0294)	0.6913 (0.0251)
	RNA	0.7691 (0.0277)	0.7520 (0.0288)	0.7193 (0.0259)
	XGBoost	0.7704 (0.0297)	0.7535 (0.0284)	0.7210 (0.0252)
	Forêt aléatoire	0.7636 (0.0299)	0.7654 (0.0282)	0.7198 (0.0253)
	LDA	0.7556 (0.0272)	0.7579 (0.0289)	0.7035 (0.0247)
CD-HIT	Régression Logistique	0.8032 (0.0135)	0.8193 (0.0144)	0.7653 (0.0128)
	RNA	0.8109 (0.0134)	0.8220 (0.0158)	0.7893 (0.0134)
	XGBoost	0.8004 (0.0137)	0.8145 (0.0154)	0.7710 (0.0152)
	Forêt aléatoire	0.7636 (0.0139)	0.7923 (0.0153)	0.7598 (0.0133)
	LDA	0.7756 (0.0132)	0.7588 (0.0199)	0.7435 (0.0274)
Aléatoire	Régression Logistique	0.8032 (0.0065)	0.7993 (0.0084)	0.7892 (0.0081)
	RNA	0.8109 (0.0064)	0.8020 (0.0078)	0.8093 (0.0089)
	XGBoost	0.8004 (0.0067)	0.7935 (0.0084)	0.7910 (0.0082)
	Forêt aléatoire	0.7636 (0.0069)	0.7654 (0.0082)	0.7698 (0.0083)
	LDA	0.7756 (0.0132)	0.7588 (0.0199)	0.7935 (0.0077)

4.1.3 Sélection de variables

Cette section compare les résultats obtenus par notre méthode originale de sélection de variables avec ceux issus de différentes approches existantes dans la littérature. Les modèles utilisés ici intègrent une étape de standardisation ainsi que notre approche originale d'échantillonnage et de balancement des classes, puisqu'il s'agit de la combinaison ayant démontré les meilleures performances jusqu'à présent.

La présentation des résultats est organisée en deux scénarios distincts. Dans le premier, l'accent est mis sur la performance du modèle en ne limitant pas le nombre de variables sélectionnées par les algorithmes. Dans le second scénario, l'accent est plutôt mis sur l'interprétabilité des résultats. Nous limitons alors le nombre de variables sélectionnées à 22, soit une variable par propriété physico-chimique (8 variables), une variable parmi celles décrivant la densité de l'environnement, ainsi que les 13 variables indiquant les types d'acides aminés adjacents. Rappelons que ce choix minimal de variables est appuyé par les résultats découlant des analyses préliminaires de corrélation et des tests statistiques réalisés lors de la présentation de la méthode originale.

Scénario de priorisation des performances

Pour obtenir les résultats des méthodes mRmR et forêt aléatoire présentés dans cette section, plusieurs seuils relatifs au nombre de variables sélectionnées ont été évalués, et les seuils ayant offert les meilleures performances ont été retenus. Ce processus est nécessaire pour ces méthodes puisqu'elles n'utilisent pas de règles précises d'inclusion ou d'exclusion des variables. À l'inverse, pour les méthodes de Mann-Whitney U et de corrélation de Spearman, nous avons directement utilisé la significativité des tests statistiques pour trancher sur l'inclusion ou l'exclusion de chaque variable. On observe que la méthode originale permet d'atteindre

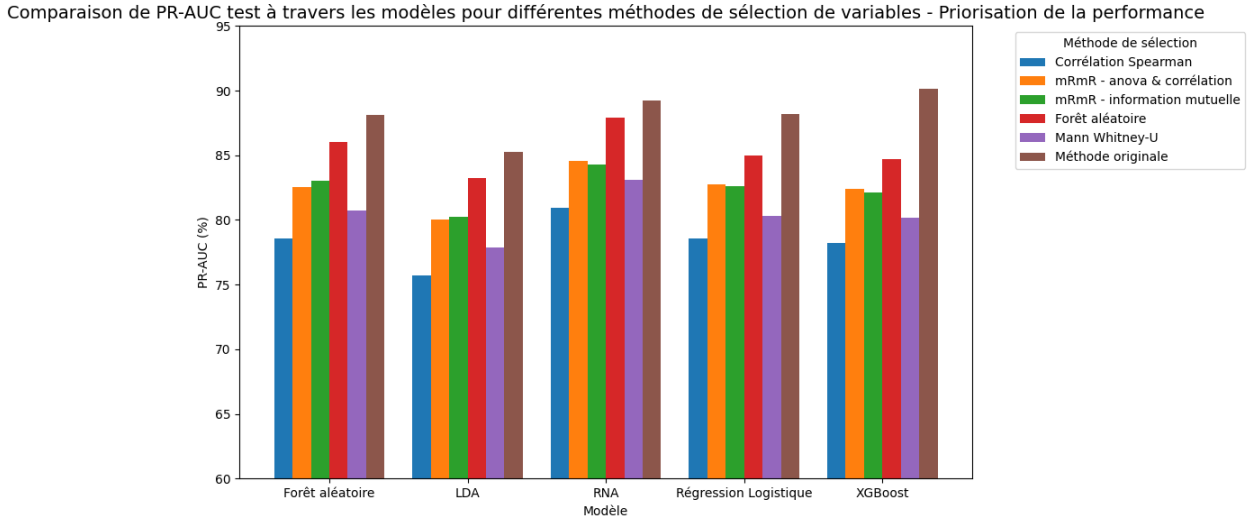

FIGURE 4.3 Test PR-AUC à travers les modèles pour différentes méthodes de sélection de variables dans un scénario de priorisation de la performance.

les meilleures performances de test pour les 5 modèles, avec un sommet de 0,9015 obtenu par XGBoost. Les approches les plus performantes sont ensuite la forêt aléatoire et l'algorithme mRmR (anova et corrélation), qui maintiennent des PR-AUC supérieures à 0,84 pour les modèles RNA, XGBoost et forêt aléatoire. À l'inverse, la corrélation de Spearman s'avère la moins efficace, son caractère purement monotone mène probablement à l'exclusion d'interactions essentielles à la tâche de prédiction. On note également dans le tableau de résultats que l'écart entre les performances d'entraînement et de test demeure modeste, suggérant que le surapprentissage est bien contrôlé.

TABLEAU 4.3 Comparaison de l'AUC-PR pour différentes méthodes de sélection de variables en entraînement, validation croisée (VC) et test, dans un scénario de maximisation des performances.

Sélection de variables	Modèle	Entraînement PR-AUC	VC PR-AUC	Test PR-AUC
Méthode originale	Régression Logistique	0.8805 (0.0030)	0.8754 (0.0032)	0.8821 (0.0038)
	RNA	0.8851 (0.0029)	**0.8993** (0.0031)	0.8926 (0.0037)
	XGBoost	**0.8907** (0.0033)	0.8895 (0.0035)	**0.9015 (0.0039)**
	Forêt aléatoire	0.8723 (0.0031)	0.8702 (0.0030)	0.8689 (0.0034)
	LDA	0.8427 (0.0042)	0.8395 (0.0040)	0.8357 (0.0045)
mRmR - anova & corrélation	Régression Logistique	0.8245 (0.0035)	0.8330 (0.0037)	0.8274 (0.0041)
	RNA	0.8418 (0.0034)	0.8399 (0.0036)	0.8454 (0.0040)
	XGBoost	0.8281 (0.0032)	0.8311 (0.0034)	0.8243 (0.0038)
	Forêt aléatoire	0.8189 (0.0036)	0.8167 (0.0035)	0.8154 (0.0039)
	LDA	0.8034 (0.0041)	0.8012 (0.0039)	0.7995 (0.0044)
mRmR - information mutuelle	Régression Logistique	0.8261 (0.0036)	0.8305 (0.0038)	0.8262 (0.0042)
	RNA	0.8493 (0.0033)	0.8422 (0.0035)	0.8427 (0.0039)
	XGBoost	0.8207 (0.0034)	0.8240 (0.0036)	0.8215 (0.0040)
	Forêt aléatoire	0.8145 (0.0032)	0.8122 (0.0034)	0.8105 (0.0038)
	LDA	0.8027 (0.0040)	0.8004 (0.0038)	0.7982 (0.0043)
Corrélation Spearman	Régression Logistique	0.7884 (0.0037)	0.7927 (0.0039)	0.7859 (0.0043)
	RNA	0.8106 (0.0035)	0.8137 (0.0037)	0.8093 (0.0041)
	XGBoost	0.7783 (0.0036)	0.7878 (0.0038)	0.7824 (0.0042)
	Forêt aléatoire	0.7709 (0.0038)	0.7685 (0.0036)	0.7662 (0.0040)
	LDA	0.7595 (0.0043)	0.7570 (0.0041)	0.7554 (0.0045)
Forêt aléatoire	Régression Logistique	0.8532 (0.0032)	0.8535 (0.0034)	0.8498 (0.0039)
	RNA	0.8806 (0.0030)	0.8839 (0.0032)	0.8791 (0.0036)
	XGBoost	0.8654 (0.0031)	0.8628 (0.0033)	0.8595 (0.0037)
	Forêt aléatoire	0.8571 (0.0033)	0.8549 (0.0032)	0.8526 (0.0036)
	LDA	0.8384 (0.0040)	0.8361 (0.0038)	0.8335 (0.0043)

Scénario de priorisation de l'interprétabilité

Ce scénario additionnel permet d'évaluer les modèles dans un contexte où l'on limite le nombre de variables au strict minimum, tout en préservant la capacité d'interpréter le profil physico-chimique (8 variables) et structurel (1 variable) de l'acide aminé étudié, en plus de conserver la capacité d'inférence des kinases potentiellement impliquées grâce aux 13 variables décrivant la fenêtre séquentielle.

On observe le même comportement qu'au scénario précédent : la méthode originale est celle permettant d'obtenir les meilleurs résultats. La méthode de sélection par forêt aléatoire est toujours celle qui semble offrir les meilleurs résultats parmi les méthodes comparatives. On note aussi que la performance maximale est encore atteinte par le modèle XGBoost, avec une valeur de 0,8712, indiquant que notre méthode permet de réduire le nombre total de variables, sans affecter de manière importante les performances (perte de 0,03 comparativement au scénario de maximisation des performances).

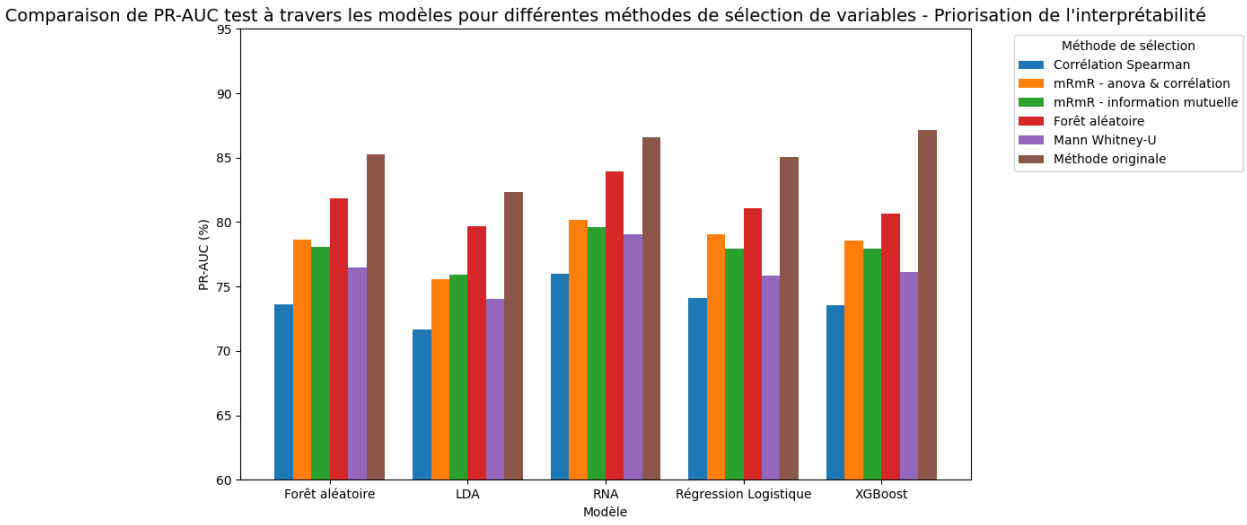

FIGURE 4.4 AUC-PR à travers les modèles pour différentes méthodes de sélection de variables avec un accent sur l'interprétabilité.

4.1.4 Comparaison avec les modèles de la littérature

3 modèles de la littérature sont utilisés dans cette section pour comparer leur performance avec notre modèle : AttenPhos [19] (séquentiel), PhosphoSVM [26] (structuro-séquentiel) ainsi que Phos-AF [31] (tridimensionnel). Ces modèles ont été sélectionnés car ils offrent les meilleures performances dans leur catégorie de modélisation respective et ce sont des modèles non spécifiques aux kinases, comme le nôtre. Afin d'augmenter la robustesse de la comparaison, les performances sont évaluées sur 3 jeux de données distincts : PELM (employé dans cette étude et celle d'AttenPhos), PhosPhAt (employé dans l'étude PhosphoSVM) et PhosphoSitePlus (employé dans l'étude Phos-AF). Puisque le modèle XGBoost s'est avéré être le plus performant jusqu'à présent, il a été retenu comme modèle final et donc pour produire les prédictions dans cette section.

Notre modèle original affiche la meilleure performance selon la PR-AUC, devant Phos-AF, Attenphos et PhosphoSVM, respectivement. La même hiérarchie ressort dans les cinq autres graphiques comparatifs (disponibles en annexe) pour les 5 autres métriques considérées.

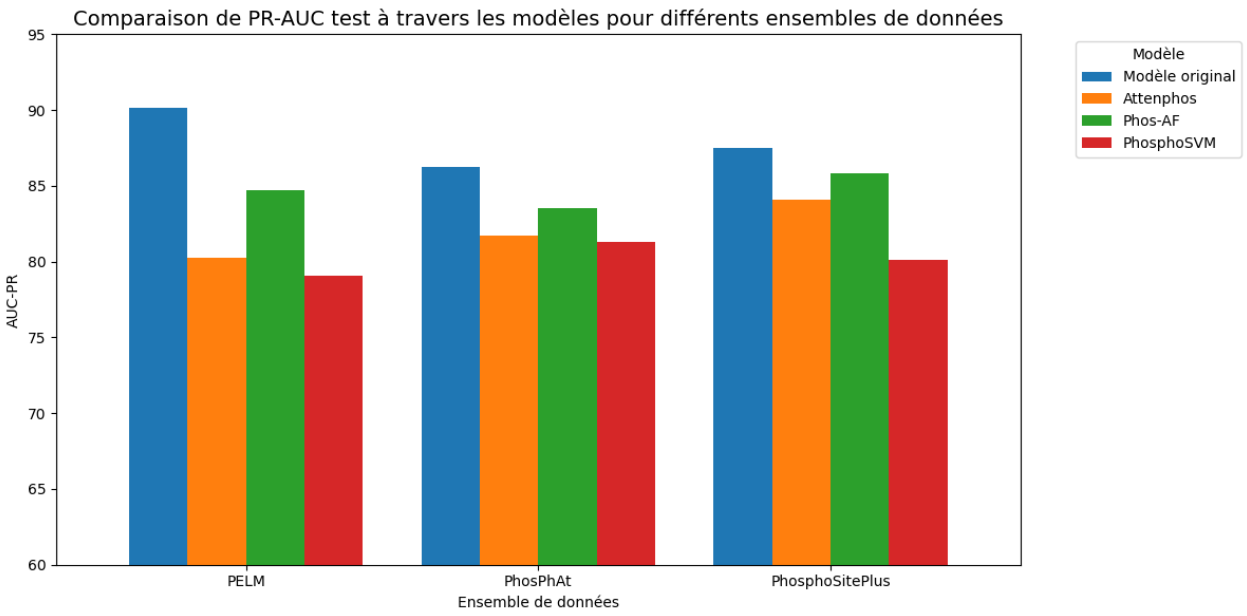

FIGURE 4.5 Comparaison de la métrique PR-AUC entre les modèles pour différents ensembles de données.

4.2 Analyse des résultats

Les résultats obtenus indiquent que nos contributions originales en modélisation, en échantillonnage et balancement des classes ainsi qu'en sélection de variables permettent d'atteindre de bonnes performances prédictives, tout en évitant l'utilisation de méthodes de style boîte noire. Le modèle proposé dans ce mémoire pourrait ainsi constituer un outil pertinent pour repérer les acides aminés les plus susceptibles d'être phosphorylés, et par conséquent, pour identifier de potentiels biomarqueurs associés aux maladies neurodégénératives.

4.2.1 Mise à l'échelle

Dans un premier temps, nous avons évalué 4 méthodes de mise à l'échelle des données. Aucune des approches testées ne s'est démarquée de manière significative. Nous avons choisi la standardisation pour les étapes subséquentes, car il s'agit d'une méthode particulièrement bien adaptée aux réseaux de neurones, que nous utilisons notamment dans l'autoencodeur intégré à notre méthode originale d'échantillonnage et de balancement des classes.

4.2.2 Échantillonnage et balancement des classes

Nous avons ensuite testé différentes méthodes pour s'attaquer au problème de débalancement des classes. Les gains de performance se sont révélés significatifs. En effet, en comparaison avec les résultats découlant de la mise à l'échelle, notre méthode originale a permis d'augmenter la PR-AUC de 4% à 8%, selon le type de modèle prédictif. De plus, parmi toutes les méthodes évaluées, y compris CD-HIT (largement employée dans la littérature) notre approche s'est avérée être celle enregistrant le meilleur gain de performance à cette étape.

Pour d'obtenir nos résultats, plusieurs configurations d'hyperparamètres propres à notre méthode originale ont été testées. Parmi ceux-ci, l'hyperparamètre $\alpha \in [0, 1]$ s'est révélé particulièrement important. Ce dernier permet de balancer l'influence relative des deux composantes utilisées dans le calcul des scores limites : la distance de Mahalanobis et les prédictions du méta-classificateur. Rappelons que les scores limites sont calculés comme suit :

$$score_{a,s} = \frac{\alpha}{M} \sum_{j=1}^{M} \hat{p}^{j}_{(a,s)} + \frac{(1-\alpha)}{D_M(\mathbf{x}_{a,s}; \mathcal{D}^{+}_{train})}, \tag{4.1}$$

et que ceux-ci visent à estimer dans quelle mesure chaque observation issue de la classe négative est susceptible d'appartenir en réalité à la classe positive. Les observations avec un score élevé sont ensuite retirées pour atténuer le bruit dans les données d'entraînement. Voici une figure qui présente le comportement du score limite de chaque observation pour différentes valeurs de α, en fonction de la probabilité du méta-classificateur et l'inverse de la distance de Mahalanobis.

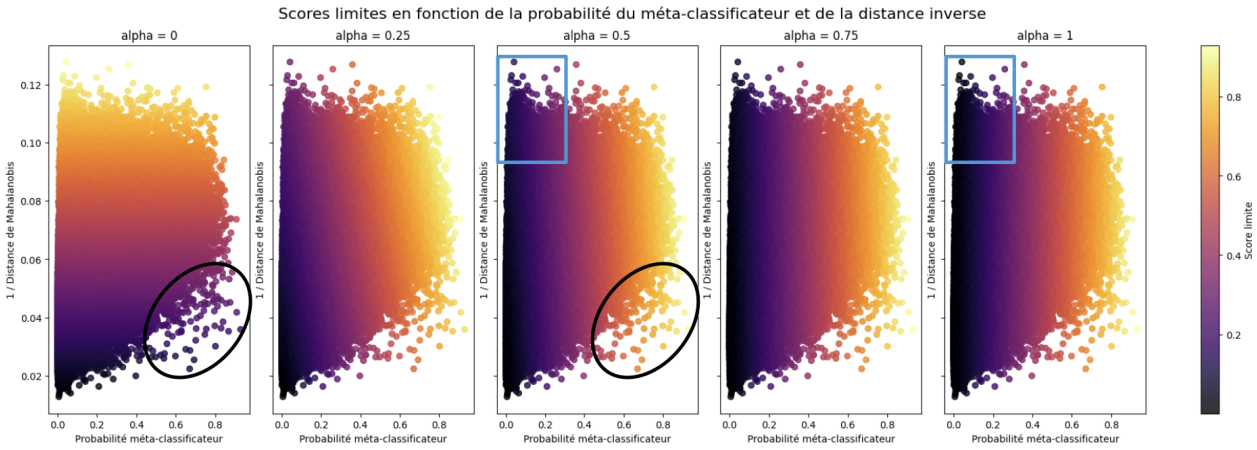

FIGURE 4.6 Comportement des scores limites en fonction de la sortie du méta-classificateur et l'inverse de la distance de Mahalanobis pour différentes valeurs de α.

Dans la figure, plus une observation est claire, plus son score limite est grand, et donc plus elle est considérée comme « bruitée » (susceptible d'appartenir réellement à la classe positive). D'abord, on observe à l'aide des cercles noirs que pour $\alpha = 0$, les observations dans le cercle sont plutôt sombres, indiquant un petit score limite (faibles chances d'appartenir à la classe positive). Cependant, on remarque que les probabilités d'appartenir à la classe positive, calculées par le méta-classificateur, sont hautes pour ces observations, ce qui est contradictoire. Lorsqu'on augmente α à 0,5, on observe que le bruit est mieux capturé pour ces observations (score limite plus élevé), ce que la distance de Mahalanobis n'a pas su faire à elle seule ($\alpha = 0$).

Inversement, dans les encadrés bleus, on constate qu'avec $\alpha = 1$, malgré une proximité à la distribution des positifs, ces observations reçoivent un score limite faible, les classant comme non bruitées, ce qui est également contradictoire. En revanche, en intégrant également la composante distance ($\alpha = 0.5$), leur score est ajusté à la hausse, rétablissant une cohérence entre la distance et la probabilité du méta-classificateur. Cette figure démontre que combiner l'information fournie par deux approches différentes (une métrique de distance et un méta-classificateur) permet d'augmenter la robustesse face aux données bruitées. Pour notre étude, cette robustesse a contribué à améliorer la qualité des données sélectionnées pour l'entraînement, menant à une augmentation de la capacité de généralisation de notre modèle. Cependant, pour être en mesure de valider la robustesse des scores limites, cela nécessiterait des validations en laboratoire sur les échantillons concernés.

4.2.3 Sélection de variables

Nous avons ensuite fait l'implémentation de différentes méthodes de sélection de variables en deux temps. Dans un premier temps, l'objectif était de trouver des solutions permettant d'augmenter les performances des modèles, ce qui a été fait avec succès. En effet, les résultats suggèrent que notre méthode originale permet d'explorer efficacement l'espace solution et de trouver des combinaisons de variables menant à une augmentation de la performance (PR-AUC en test atteinte de 0,9015 versus sommet de 0,8568 sans méthode de sélection de variables). Aussi, la comparaison des résultats avec d'autres méthodes employées dans la littérature indique que notre méthode permet d'atteindre une meilleure capacité de généralisation des modèles prédictifs qu'à l'aide des méthodes actuellement employées dans le domaine.

Dans un deuxième temps, nous avons testé les performances de notre modèle dans un scénario où l'utilisateur désire restreindre le plus possible le nombre de variables incluses (22 variables), afin de simplifier l'interprétation des résultats. Les résultats de cette section ont montré que

notre méthode de sélection de variables permet de former des solutions de combinaison de variables de petite taille, tout en limitant l'impact sur la capacité de généralisation. En effet, un sommet de la PR-AUC en test de 0,8712 a été atteint, soit une réduction de la performance de 0,03 comparativement à un scénario où le nombre de variables n'est pas restreint (PR-AUC 0,9015).

4.2.4 Comparaison avec les modèles de la littérature

Enfin, nous avons évalué le modèle optimal identifié (XGBoost) par rapport à trois autres approches de la littérature, sur trois ensembles de données distincts. XGBoost obtient les meilleures performances pour chacune des six métriques retenues, et ce sur l'ensemble des trois jeux de données. Phos-AF se classe en deuxième position, avec un écart de performance compris entre 2 % et 5 % selon la métrique considérée, tandis qu'AttenPhos et PhosphoSVM présentent des écarts similaires, situés entre 5 % et 10 %. Ces résultats sont en lien avec nos attentes, car Phos-AF est le seul modèle qui intègre des informations tridimensionnelles, un aspect dont nous avons souligné l'importance pour la modélisation des sites de phosphorylation. On peut donc conclure que notre modèle original offre des performances lui permettant d'être employé pour identifier des acides aminés susceptibles d'être phosphorylés.

CHAPITRE 5 APPLICATION DIRECTE DU MODÈLE : IDENTIFICATION DE BIOMARQUEURS POTENTIELS DE LA MALADIE DE HUNTINGTON

5.1 Introduction

L'objectif de ce chapitre est d'identifier un groupe d'acides aminés appartenant à la protéine tau ayant le potentiel de servir de biomarqueurs pour la maladie de Huntington. Pour ce faire, nous allons employer le modèle de prédiction de la phosphorylation développé précédemment. Notre motivation à développer un modèle prédictif découle en partie de notre contribution à une étude scientifique [55], portant sur la quantification expérimentale de la phosphorylation de la protéine tau. Notre participation à cette étude est ce qui nous a permis de constater la complexité, la durée et le coût importants associés à ces analyses. C'est donc cette observation qui a motivé notre choix de développer un modèle prédictif permettant de cibler prioritairement les résidus les plus susceptibles d'être phosphorylés.

Rappelons que la protéine tau est composée de 79 sérines, 50 thréonines et 6 tyrosines et que la phosphorylation n'a été confirmée expérimentalement que sur 26, 13 et 4 de ces résidus, respectivement. Cela laisse donc un total de 92 acides aminés pour lesquels la phosphorylation n'est toujours pas validée. Dans cette section, nos efforts se concentreront sur 42 résidus spécifiques (20 sérines, 21 thréonines et 1 tyrosine). Ces résidus ont été sélectionnés parce qu'ils sont présents dans les variantes de la protéine tau exprimées dans le cerveau humain, contrairement aux 40 autres résidus qui se trouvent uniquement dans des variantes plus rares et absentes du système nerveux central.

5.2 Méthodologie

Afin d'obtenir les résultats, nous proposons d'agréger les prédictions issues des 5 types de modèles prédictifs employés dans ce mémoire, puis de générer des intervalles de confiance à partir d'un total de 1 000 prédictions (250 par modèle) pour chacun des 42 acides aminés ciblés.

Intervalle de confiance avec correction du biais et de l'accélération

Pour produire les intervalles de confiance, nous employons la méthode avec correction du biais et de l'accélération (BCA), qui est semblable à l'intervalle des quantiles, sauf qu'au lieu

de choisir des quantiles fixes, la méthode BCA choisit différents quantiles en tenant compte du biais et de l'asymétrie de la distribution de nos prédictions. Les bornes de l'intervalle de confiance s'expriment comme suit :

$$\Big[L_{\mathrm{BCa}},\, U_{\mathrm{BCa}}\Big] = \Big[q^*_{\alpha_{\mathrm{inf}}},\, q^*_{\alpha_{\mathrm{sup}}}\Big], \tag{5.1}$$

où q^*_γ désigne le γ-quantile empirique de la distribution bootstrap $\{T^*_1, \ldots, T^*_B\}$. α_{inf} et α_{sup} tiennent compte à la fois du biais médian et de l'asymétrie de la statistique et s'expriment comme suit :

$$\alpha_{\mathrm{inf}} = \Phi\Big(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a\,(z_0 + z_{\alpha/2})}\Big), \qquad \alpha_{\mathrm{sup}} = \Phi\Big(z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a\,(z_0 + z_{1-\alpha/2})}\Big), \tag{5.2}$$

où Φ est la fonction de répartition de $\mathcal{N}(0,1)$ et $z_p = \Phi^{-1}(p)$. Cette approche est particulièrement adaptée à notre contexte, car l'agrégation des 1 000 prédictions issues de 5 modèles différents risque d'engendrer des distributions biaisées et asymétriques, que les ajustements du biais et de l'accélération corrigent directement.

5.3 Résultats

Voici un tableau montrant les intervalles de confiance des 5 acides aminés de la protéine tau pour lesquels les probabilités de phosphorylation prédites sont les plus élevées. Le tableau présentant les intervalles de confiance pour l'ensemble des 42 acides aminés ciblés est disponible en annexe.

Tableau 5.1 Intervalles de confiance à 95 % pour les cinq résidus de la protéine tau présentant les probabilités de phosphorylation les plus élevées.

Type d'acide aminé	**Position**	**Intervalle de confiance**
Thréonine	52	[0,9034 ; 0,9565]
Sérine	137	[0,8804 ; 0,9341]
Sérine	56	[0,8660 ; 0,9071]
Thréonine	361	[0,8496 ; 0,8955]
Thréonine	123	[0,8435 ; 0,8785]

5.4 Discussion des résultats

Les prédictions issues de notre méthode révèlent une certaine hiérarchie parmi les 42 sites de phosphorylation potentiels. Nous avons ainsi classé les résidus ciblés en 3 groupes, selon leur potentiel à servir de biomarqueurs.

1. **Potentiel élevé :** Ces 5 résidus présentent des intervalles de confiance avec des bornes inférieures plus grandes ou égales à 0,8435.
2. **Potentiel modéré :** Ce groupe est composé de 11 acides aminés avec des bornes inférieures comprises entre 0,5282 et 0,7627.
3. **Potentiel faible :** 26 résidus se retrouvent dans ce groupe. Les bornes inférieures de leurs intervalles de confiance ne dépassent pas 0,3977.

Un tableau fourni en annexe permet de lier chaque acide aminé à son groupe respectif. Une autre observation intéressante concerne la nature des acides aminés figurant dans le groupe à potentiel élevé ou le groupe à potentiel modéré. Parmi ces 16 résidus, on dénombre 11 thréonines (69 %), alors que les thréonines ne représentent que 50 % des 42 résidus étudiés. Cette prédominance pourrait refléter une préférence des kinases actives dans le contexte pathologique de la maladie de Huntington, un motif que le modèle semble avoir capturé, et qui pourrait être une piste intéressante pour les chercheurs.

En conclusion, nous avons été en mesure d'établir une liste de résidus prioritaires pour mieux orienter les futures études d'identification de biomarqueurs. D'autre part, notre méthode nous a également permis d'identifier un groupe d'acides aminés pour lesquels la phosphorylation est peu probable, suggérant qu'il serait judicieux de les exclure des analyses en laboratoire, ou du moins, d'en faire des cibles moins prioritaires.

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

Ce mémoire avait pour objectif de développer un modèle de prédiction de la phosphorylation d'acides aminés à la fois plus performant et interprétable que les modèles actuels, afin d'identifier de potentiels biomarqueurs liés aux maladies neurodégénératives. Pour ce faire, nous avons adopté une approche systématique d'apprentissage machine, dans laquelle se sont insérées trois contributions originales visant à surmonter des limitations identifiées au sein des modèles de la littérature.

La première contribution a porté sur la modélisation des acides aminés. Plus spécifiquement, nous avons proposé une méthode de caractérisation spatiale plus robuste, qui se sert du centre de masse de chaque acide aminé et qui agrège les propriétés physico-chimiques des résidus voisins au sein de six sphères de rayons croissants. Ainsi, nous avons pu générer un ensemble riche de variables modélisant l'environnement spatial. La deuxième contribution a consisté à développer une méthode d'échantillonnage et de balancement des classes. Dans notre contexte, les jeux de données disponibles souffrent d'un fort déséquilibre et d'une incertitude liée à la classe des résidus étiquetés comme négatifs. Pour adresser ces problèmes, nous avons d'abord attribué à chaque observation négative un score de confiance, ce qui a permis de filtrer les observations négatives les plus susceptibles d'être mal étiquetées. Par la suite, nous avons projeté les observations négatives fiables dans un espace latent via un auto-encodeur, puis appliqué un algorithme de k-moyennes suivi d'un sous-échantillonnage par k-centres pour obtenir un ensemble d'entraînement négatif à la fois représentatif, diversifié et moins bruité. Notre troisième contribution était le développement d'une méthode hybride de sélection de variables qui explore l'espace des solutions en générant itérativement des sous-ensembles de variables de manière probabiliste. La probabilité de sélection de chaque variable était guidée par un score de pertinence a priori et modulée par un hyperparamètre de température.

L'intégration de ces contributions à l'approche systématique d'apprentissage machine nous a permis d'obtenir un modèle optimisé qui a atteint une valeur de PR-AUC de 0,9015 en phase de test. Nous avons ensuite comparé notre méthode avec d'autres modèles de la littérature et avons constaté que les performances atteintes par notre modèle étaient supérieures, et ce, sur l'ensemble des métriques de performance considérées.

Dans le deuxième chapitre, nous avons directement appliqué notre modèle sur 42 acides

aminés de la protéine tau, dans le but d'identifier de potentiels biomarqueurs de la maladie de Huntington. Notre choix d'appliquer notre modèle à cette protéine et cette maladie a été motivé par le constat que nous avons fait lors d'une contribution à une étude visant à identifier des biomarqueurs de la maladie de Huntington via la quantification de la phosphorylation sur différents acides aminés de la protéine tau. En effet, nous avons constaté à quel point il était complexe et coûteux de réaliser ce genre d'expériences en laboratoire. Après avoir mis en application notre modèle, nous avons identifié et hiérarchisé les 42 résidus d'intérêt, mettant en évidence un groupe de 5 acides aminés ayant un fort potentiel de servir de biomarqueur. Cette application a permis de démontrer la pertinence de notre approche pour générer des hypothèses ciblées, optimisant ainsi les futurs efforts de recherche expérimentale.

6.2 Limitations de l'approche

Bien que l'approche développée dans ce mémoire ait démontré de bonnes performances, une analyse critique révèle plusieurs limitations inhérentes aux choix méthodologiques effectués.

La première limitation de notre méthode concerne la nature même des données spatiales utilisées. Notre méthode de modélisation tridimensionnelle s'appuie sur les structures prédites par AlphaFold. Or, ces données demeurent des prédictions et elles sont donc sujettes à de l'incertitude. Également, ces coordonnées spatiales ne reflètent qu'une conformation statique de la protéine d'intérêt. Toutefois, dans un environnement biologique, les protéines sont des entités dynamiques qui fluctuent et changent de conformation, ce qui peut influencer l'accessibilité des sites de phosphorylation et, par conséquent, affecter l'estimation des probabilités de phosphorylation.

La deuxième limitation identifiée est liée à notre méthode d'échantillonnage. Le calcul du score de confiance pour les observations de la classe négative dépend directement des performances du méta-classificateur initial et de la robustesse de la distance de Mahalanobis. Un méta-classificateur initial peu performant pourrait biaiser l'estimation des scores et conduire à une mauvaise identification des données bruitées.

Enfin, notre méthode de sélection de variables étant une approche heuristique, elle explore l'espace des solutions de manière intelligente mais n'offre aucune garantie de converger vers le véritable optimum global. Il est donc possible que des combinaisons de variables encore plus performantes existent mais n'aient pas été découvertes lors du processus d'exploration. De plus, la méthode introduit son propre ensemble d'hyperparamètres, tels que la température initiale, la température finale et le taux de refroidissement, dont le réglage optimal peut s'avérer complexe.

6.3 Améliorations futures

Les limitations identifiées dans la section précédente ouvrent de nombreuses perspectives d'améliorations futures et d'extensions de ce travail. Ces pistes de recherche visent à renforcer la robustesse du modèle et à approfondir sa pertinence biologique.

Pour adresser la limitation des structures protéiques statiques, une amélioration importante consisterait à intégrer la dynamique des protéines dans la modélisation. Plutôt que de se baser sur une seule structure prédite par protéine, il serait pertinent d'employer des simulations de dynamique moléculaire pour générer un ensemble de conformations structurelles représentatives de la flexibilité pour chaque protéine dans l'ensemble de données. L'analyse des sites de phosphorylation serait alors effectuée sur cet ensemble de structures, permettant ainsi de capturer des motifs dépendants de la conformation des protéines.

Concernant notre méthode d'échantillonnage et de balancement des classes, il serait pertinent d'étudier plus en profondeur les observations négatives exclues de l'entraînement car elles ont de grandes chances d'appartenir à la classe positive. Plus spécifiquement, des analyses de regroupement pourraient être effectuées afin de détecter des motifs dans ces observations et mieux comprendre pourquoi elles ont été exclues de la phase d'apprentissage.

Enfin, la méthode de sélection de variables pourrait également être perfectionnée. Des techniques d'optimisation plus avancées, telles que l'optimisation bayésienne, pourraient être utilisées pour rechercher de manière plus efficace l'espace des hyperparamètres de notre algorithme de sélection. Cela permettrait d'automatiser le processus de recherche de la meilleure solution et de trouver des combinaisons de variables encore plus performantes.

RÉFÉRENCES

[1] World Health Organization, "Dementia," 2022, available at : https://www.who.int/news-room/fact-sheets/detail/dementia.

[2] Parkinson's Foundation, "Parkinson's disease statistics," 2022, available at : https://www.parkinson.org/Understanding-Parkinsons/Statistics.

[3] C. C. for Economic Analysis, "Dementia in canada : Economic burden 2020 to 2050," Canadian Centre for Economic Analysis, Rapport technique, July 2023, accessed : 2025-03-14. [En ligne]. Disponible : https://www.cancea.ca/wp-content/uploads/2023/07/CANCEA-Economic-Impact-of-Dementia-in-Canada-2023-01-08.pdf

[4] World Health Organization, "Amyotrophic lateral sclerosis (als)," 2022, available at : https://www.who.int/news-room/fact-sheets/detail/amyotrophic-lateral-sclerosis.

[5] M. Gratuze, G. Cisbani, F. Cicchetti et E. Planel, "Is huntington's disease a tauopathy ?" *Brain*, vol. 139, n°. 4, p. 1014–1025, 03 2016. [En ligne]. Disponible : https://doi.org/10.1093/brain/aww021

[6] A. Maxan et F. Cicchetti, "Tau : A common denominator and therapeutic target for neurodegenerative disorders," *Journal of Experimental Neuroscience*, vol. 12, p. 1179069518772380, 2018, pMID : 29760562. [En ligne]. Disponible : https://doi.org/10.1177/1179069518772380

[7] U. Consortium, "Microtubule-associated protein tau," https://www.uniprot.org/uniprotkb/P10636/entry, accessed : 2025-03-14.

[8] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts et P. Walter, *The shape and structure of proteins*. London, England : Garland Science, 2002.

[9] E. A. Ponomarenko, E. V. Poverennaya, E. V. Ilgisonis, M. A. Pyatnitskiy, A. T. Kopylov, V. G. Zgoda, A. V. Lisitsa et A. I. Archakov, "The size of the human proteome : The width and depth," *Int J Anal Chem*, vol. 2016, p. 7436849, mai 2016.

[10] F. Esmaili, M. Pourmirzaei, S. Ramazi, S. Shojaeilangari et E. Yavari, "A review of machine learning and algorithmic methods for protein phosphorylation site prediction," *Genomics Proteomics Bioinformatics*, vol. 21, n°. 6, p. 1266–1285, oct. 2023.

[11] T. H. Dang, K. Van Leemput, A. Verschoren et K. Laukens, "Prediction of kinase-specific phosphorylation sites using conditional random fields," *Bioinformatics*, vol. 24, n°. 24, p. 2857–2864, 10 2008. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/btn546

[12] Z. Zhou, W. Yeung, N. Gravel, M. Salcedo, S. Soleymani, S. Li et N. Kannan, “Phosformer : an explainable transformer model for protein kinase-specific phosphorylation predictions,” *Bioinformatics*, vol. 39, n°. 2, p. btad046, 01 2023. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/btad046

[13] P. Shrestha, J. Kandel, H. Tayara et K. T. Chong, “Dl-sphos : Prediction of serine phosphorylation sites using transformer language model,” *Computers in Biology and Medicine*, vol. 169, p. 107925, 2024. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S001048252400009X

[14] A. K. Biswas, N. Noman et A. R. Sikder, “Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information,” *BMC Bioinformatics*, vol. 11, n°. 1, p. 273, May 2010. [En ligne]. Disponible : https://doi.org/10.1186/1471-2105-11-273

[15] S. Basu et D. Plewczynski, “Ams 3.0 : prediction of post-translational modifications,” *BMC Bioinformatics*, vol. 11, n°. 1, p. 210, Apr 2010. [En ligne]. Disponible : https://doi.org/10.1186/1471-2105-11-210

[16] H. D. Ismail, A. Jones, J. H. Kim, R. H. Newman et D. B. KC, “Rf-phos : A novel general phosphorylation site prediction tool based on random forest,” *BioMed Research International*, vol. 2016, n°. 1, p. 3281590, 2016. [En ligne]. Disponible : https://onlinelibrary.wiley.com/doi/abs/10.1155/2016/3281590

[17] L. Wei, P. Xing, J. Tang et Q. Zou, “Phospred-rf : A novel sequence-based predictor for phosphorylation sites using sequential information only,” *IEEE Transactions on Nano-Bioscience*, vol. 16, n°. 4, p. 240–247, 2017.

[18] X. Wang, Z. Zhang, C. Zhang, X. Meng, X. Shi et P. Qu, “Transphos : A deep-learning model for general phosphorylation site prediction based on transformer-encoder architecture,” *International Journal of Molecular Sciences*, vol. 23, n°. 8, 2022. [En ligne]. Disponible : https://www.mdpi.com/1422-0067/23/8/4263

[19] T. Song, Q. Yang, P. Qu, L. Qiao et X. Wang, “Attenphos : General phosphorylation site prediction model based on attention mechanism,” *International Journal of Molecular Sciences*, vol. 25, n°. 3, 2024. [En ligne]. Disponible : https://www.mdpi.com/1422-0067/25/3/1526

[20] S. Ahmed, M. Kabir, M. Arif, Z. U. Khan et D.-J. Yu, “Deepppsite : A deep learning-based model for analysis and prediction of phosphorylation sites using efficient sequence information,” *Analytical Biochemistry*, vol. 612, p. 113955, 2021. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0003269720304875

[21] J. Gao, J. J. Thelen, A. K. Dunker et D. Xu, “Musite, a tool for global prediction of general and kinase-specific phosphorylation sites *,” *Molecular &*

Cellular Proteomics, vol. 9, nº. 12, p. 2586–2600, Dec 2010. [En ligne]. Disponible : https://doi.org/10.1074/mcp.M110.001388

[22] D. Wang, S. Zeng, C. Xu, W. Qiu, Y. Liang, T. Joshi et D. Xu, "Musitedeep : a deep-learning framework for general and kinase-specific phosphorylation site prediction," *Bioinformatics*, vol. 33, nº. 24, p. 3909–3916, 08 2017. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/btx496

[23] F. Luo, M. Wang, Y. Liu, X.-M. Zhao et A. Li, "Deepphos : prediction of protein phosphorylation sites with deep learning," *Bioinformatics*, vol. 35, nº. 16, p. 2766–2773, 01 2019. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/bty1051

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser et I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [En ligne]. Disponible : http://arxiv.org/abs/1706.03762

[25] W.-R. Qiu, X. Xiao, Z.-C. Xu et K.-C. Chou, "iphos-pseen : Identifying phosphorylation sites in proteins by fusing different pseudo components into an ensemble classifier," *Oncotarget*, vol. 7, nº. 32, p. 51 270–51 283, 2016. [En ligne]. Disponible : https://www.oncotarget.com/article/9987/

[26] Y. Dou, B. Yao et C. Zhang, "Phosphosvm : prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine," *Amino Acids*, vol. 46, nº. 6, p. 1459–1469, Jun 2014. [En ligne]. Disponible : https://doi.org/10.1007/s00726-014-1711-5

[27] J. J. Ward, L. J. McGuffin, K. Bryson, B. F. Buxton et D. T. Jones, "The disopred server for the prediction of protein disorder," *Bioinformatics*, vol. 20, nº. 13, p. 2138–2139, 03 2004. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/bth195

[28] D. M. N. Kumar, N. Prakash, "Getting phosphorylated : Is it necessary to be solvent accessible ?" *Proceedings of the Indian National Science Academy*, vol. 81, 03 2015.

[29] L. J. McGuffin, K. Bryson et D. T. Jones, "The psipred protein structure prediction server," *Bioinformatics*, vol. 16, nº. 4, p. 404–405, 04 2000. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/16.4.404

[30] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli et D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, nº. 7873, p. 583–589, Aug 2021. [En ligne]. Disponible : https://doi.org/10.1038/s41586-021-03819-2

[31] Z. Yu, J. Yu, H. Wang, S. Zhang, L. Zhao et S. Shi, "Phosaf : An integrated deep learning architecture for predicting protein phosphorylation sites with alphafold2 predicted structures," *Analytical Biochemistry*, vol. 690, p. 115510, 2024. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S000326972400054X

[32] W. Li et A. Godzik, "Cd-hit : a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, nº. 13, p. 1658–1659, 05 2006. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/btl158

[33] S. F. Altschul, W. Gish, W. Miller, E. W. Myers et D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, nº. 3, p. 403–410, oct. 1990.

[34] J. H. Kim, J. Lee, B. Oh, K. Kimm et I. Koh, "Prediction of phosphorylation sites using svms," *Bioinformatics*, vol. 20, nº. 17, p. 3179–3184, 07 2004. [En ligne]. Disponible : https://doi.org/10.1093/bioinformatics/bth382

[35] C. Strobl, A.-L. Boulesteix, A. Zeileis et T. Hothorn, "Bias in random forest variable importance measures : Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, nº. 1, p. 25, Jan 2007. [En ligne]. Disponible : https://doi.org/10.1186/1471-2105-8-25

[36] C.-W. Chen, L.-Y. Huang, C.-F. Liao, K.-P. Chang et Y.-W. Chu, "Gasphos : Protein phosphorylation site prediction using a new feature selection approach with a ga-aided ant colony system," *International Journal of Molecular Sciences*, vol. 21, nº. 21, 2020. [En ligne]. Disponible : https://www.mdpi.com/1422-0067/21/21/7891

[37] K. S. Bhullar, N. O. Lagarón, E. M. McGowan, I. Parmar, A. Jha, B. P. Hubbard et H. P. V. Rupasinghe, "Kinase-targeted cancer therapies : progress, challenges and future directions," *Molecular Cancer*, vol. 17, nº. 1, p. 48, Feb 2018. [En ligne]. Disponible : https://doi.org/10.1186/s12943-018-0804-2

[38] M. Chen, W. Zhang, Y. Gou, D. Xu, Y. Wei, D. Liu, C. Han, X. Huang, C. Li, W. Ning, D. Peng et Y. Xue, "Gps 6.0 : an updated server for prediction of kinase-specific phosphorylation sites in proteins," *Nucleic Acids Research*, vol. 51, nº. W1, p. W243–W250, 05 2023. [En ligne]. Disponible : https://doi.org/10.1093/nar/gkad383

[39] H. Dinkel, C. Chica, A. Via, C. M. Gould, L. J. Jensen, T. J. Gibson et F. Diella, "Phospho.ELM : a database of phosphorylation sites–update 2011," *Nucleic Acids Res*, vol. 39, nº. Database issue, p. D261–7, nov. 2010.

[40] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. Shoemaker, P. Thiessen, B. Yu, L. Zaslavsky, J. Zhang et E. Bolton, "Pubchem 2025 update," *Nucleic Acids Research*, vol. 53, nº. D1, p. D1516–D1525, 11 2024. [En ligne]. Disponible : https://doi.org/10.1093/nar/gkae1059

[41] M. Malumbres, "Cyclin-dependent kinases," *Genome Biology*, vol. 15, nº. 6, p. 122, Jun 2014. [En ligne]. Disponible : https://doi.org/10.1186/gb4184

[42] T. Cheng, Y. Zhao, X. Li, F. Lin, Y. Xu, X. Zhang, Y. Li, R. Wang et L. Lai, "Computation of octanolwater partition coefficients by guiding an additive model with knowledge," *Journal of Chemical Information and Modeling*, vol. 47, n°. 6, p. 2140–2148, 2007, pMID : 17985865. [En ligne]. Disponible : https://doi.org/10.1021/ci700257y

[43] E. He, G. Yan, J. Zhang, J. Wang et W. Li, "Effects of phosphorylation on the intrinsic propensity of backbone conformations of serine/threonine," *J Biol Phys*, vol. 42, n°. 2, p. 247–258, janv. 2016.

[44] S. Miller, J. Janin, A. M. Lesk et C. Chothia, "Interior and surface of monomeric proteins," *J Mol Biol*, vol. 196, n°. 3, p. 641–656, août 1987.

[45] P. Ertl, B. Rohde et P. Selzer, "Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties," *J Med Chem*, vol. 43, n°. 20, p. 3714–3717, oct. 2000.

[46] A. J. Doig, "Frozen, but no accident – why the 20 standard amino acids were selected," *The FEBS Journal*, vol. 284, n°. 9, p. 1296–1305, 2017. [En ligne]. Disponible : https://febs.onlinelibrary.wiley.com/doi/abs/10.1111/febs.13982

[47] J. B. Ahrens, J. Nunez-Castilla et J. Siltberg-Liberles, "Evolution of intrinsic disorder in eukaryotic proteins," *Cellular and Molecular Life Sciences*, vol. 74, n°. 17, p. 3163–3174, Sep 2017. [En ligne]. Disponible : https://doi.org/10.1007/s00018-017-2559-0

[48] L. M. Iakoucheva, P. Radivojac, C. J. Brown, T. R. O'Connor, J. G. Sikes, Z. Obradovic et A. K. Dunker, "The importance of intrinsic disorder for protein phosphorylation," *Nucleic Acids Res*, vol. 32, n°. 3, p. 1037–1049, févr. 2004.

[49] J. B. Hendrickson, P. Huang et A. G. Toczko, "Molecular complexity : a simplified formula adapted to individual atoms," *Journal of Chemical Information and Computer Sciences*, vol. 27, n°. 2, p. 63–67, May 1987. [En ligne]. Disponible : https://doi.org/10.1021/ci00054a004

[50] H. P. Erickson, "Size and shape of protein molecules at the nanometer level determined by sedimentation, gel filtration, and electron microscopy," *Biological Procedures Online*, vol. 11, n°. 1, p. 32, May 2009. [En ligne]. Disponible : https://doi.org/10.1007/s12575-009-9008-x

[51] C. C. Aggarwal, A. Hinneburg et D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," dans *Database Theory — ICDT 2001*, J. Van den Bussche et V. Vianu, édit. Berlin, Heidelberg : Springer Berlin Heidelberg, 2001, p. 420–434.

[52] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, p. 293–306, 1985. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/0304397585902245

[53] W. Noble, D. P. Hanger, C. C. J. Miller et S. Lovestone, "The importance of tau phosphorylation for neurodegenerative diseases," *Front Neurol*, vol. 4, p. 83, juill. 2013.

[54] R. A. Fisher, *Statistical Methods for Research Workers*, 1er éd. Edinburgh : Oliver & Boyd, 1925, accessed 10May2025. [En ligne]. Disponible : https://archive.org/details/in.ernet.dli.2015.205971

[55] M. Alpaugh, J. Lantero-Rodriguez, A. L. Benedet, U. Manseau, M. Boutin, M. Maiuri, H. L. Denis, M. Masnata, S. V. Fazal, S. Chouinard, P. Rosa-Neto, R. A. Barker, K. Blennow, H. Zetterberg, R. Labib et F. Cicchetti, "Tau levels in platelets isolated from huntington's disease patients serve as a biomarker of disease severity," *Journal of Neurology*, vol. 272, no. 3, p. 254, Mar 2025. [En ligne]. Disponible : https://doi.org/10.1007/s00415-025-12966-9

ANNEXE A GAINS MARGINAUX DE PERFORMANCE DU MODÈLE D'ANALYSE PAR DISCRIMINANT LINÉAIRE EN FONCTION DE DIFFÉRENTES TAILLES DE FENÊTRE K

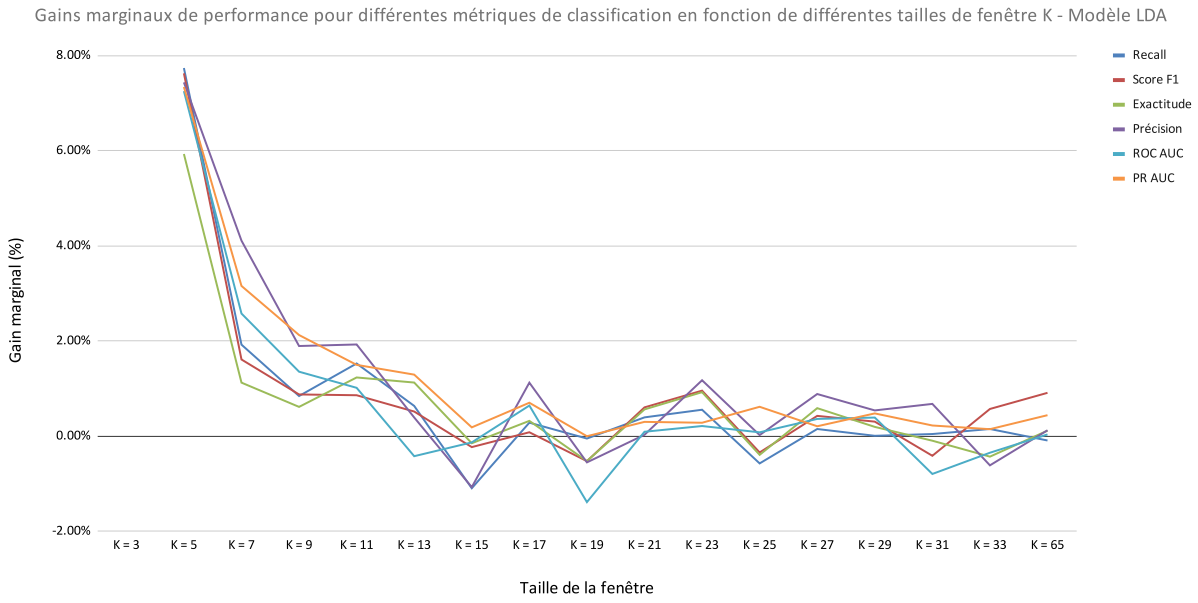

FIGURE A.1 Gains marginaux de performance du modèle d'analyse par discriminant linéaire en fonction de différentes tailles de fenêtre K.

ANNEXE B THÉORÈME DE SINGULARITÉ DE LA MATRICE DE COVARIANCE EMPLOYÉE DANS LA DISTANCE DE MAHALANOBIS $D_M(\mathbf{x_{a,s}}; \mathcal{D}^+_{train})$

Théorème

Soit $\mathbf{X}$ une matrice de données de taille $N \times p$ représentant N observations de p variables, $\mathbf{X} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T]^T$, où $\mathbf{x}_i \in \mathbb{R}^p$. Supposons qu'un sous-ensemble de k variables, indexées par l'ensemble $J = \{j_1, j_2, \ldots, j_k\} \subseteq \{1, \ldots, p\}$ avec $k \geq 2$, représente un encodage one-hot d'une unique variable discrète à k modalités. Ceci implique que, pour chaque observation $i \in \{1, \ldots, N\}$, la contrainte suivante est vérifiée :

$$\sum_{j \in J} x_{i,j} = 1. \tag{B.1}$$

Soit $\hat{\mathbf{S}}$ la matrice de covariance empirique de $\mathbf{X}$, calculée comme suit :

$$\hat{\mathbf{S}} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T, \tag{B.2}$$

où $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$ est le vecteur de moyenne empirique. Alors, la matrice $\hat{\mathbf{S}}$ est singulière.

Preuve

Une matrice carrée $\mathbf{A} \in \mathbb{R}^{p \times p}$ est singulière si et seulement s'il existe un vecteur $\mathbf{x} : \mathbf{Ax} = \mathbf{0}_p$, où $\mathbf{x} \in \mathbb{R}^p$ et $\mathbf{x} \neq \mathbf{0}_p$. Ainsi, pour montrer la singularité de la matrice de covariance carrée $\hat{\mathbf{S}} \in \mathbb{R}^{p \times p}$, nous allons prouver l'existence d'un vecteur non nul $\mathbf{v} : \hat{\boldsymbol{S}}\mathbf{v} = \mathbf{0}_p$.

Définissons d'abord $\hat{\mathbf{S}}$, en fonction de $\tilde{\mathbf{X}}$, la matrice de données centrée de taille $N \times p$, dont l'élément à la i-ème ligne et j-ème colonne est donné par $\tilde{X}_{ij} = x_{i,j} - \hat{\mu}_j$. Ainsi, la matrice de covariance empirique $\hat{\mathbf{S}}$ peut s'exprimer comme suit :

$$\hat{\mathbf{S}} = \frac{1}{N-1} \tilde{\mathbf{X}}^T \tilde{\mathbf{X}}. \tag{B.3}$$

Considérons la somme des éléments de la i-ème ligne de $\tilde{\mathbf{X}}$ correspondant aux indices dans J :

$$\begin{aligned}\sum_{j\in J}\tilde{X}_{ij} &= \sum_{j\in J}(x_{i,j}-\hat{\mu}_j)\\ &= \left(\sum_{j\in J}x_{i,j}\right)-\left(\sum_{j\in J}\hat{\mu}_j\right)\end{aligned} \tag{B.4}$$

Par la contrainte $B.1$, $\sum_{j\in J}x_{i,j}=1\ \forall\ i$. Évaluons maintenant la somme des moyennes pour les variables dans J :

$$\begin{aligned}\sum_{j\in J}\hat{\mu}_j &= \sum_{j\in J}\left(\frac{1}{N}\sum_{i=1}^{N}x_{i,j}\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j\in J}x_{i,j}\right)\quad\text{(par linéarité de la sommation)}\\ &= \frac{1}{N}\sum_{i=1}^{N}(1)\quad\text{(en appliquant la contrainte } B.1)\\ &= \frac{1}{N}(N)=1\end{aligned} \tag{B.5}$$

Substituons les résultats $\sum_{j\in J}x_{i,j}=1$ et $\sum_{j\in J}\hat{\mu}_j=1$ dans l'équation $B.4$:

$$\sum_{j\in J}\tilde{X}_{ij}=1-1=0. \tag{B.4}$$

Il y a donc une dépendance linéaire entre les colonnes de la matrice de données centrée $\tilde{\mathbf{X}}$ correspondant aux indices dans J. Spécifiquement, la somme de ces colonnes est le vecteur nul dans $\mathbb{R}^N$.

Définissons maintenant un vecteur non nul $\mathbf{v}\in\mathbb{R}^p$ dans le noyau de $\hat{\mathbf{S}}$ tel que :

$$v_j=\begin{cases}1 & \text{si } j\in J\\ 0 & \text{si } j\notin J.\end{cases}$$

$\mathbf{v}$ est bel et bien un vecteur non nul, car J est non vide, puisqu'il contient k indices et $k\geq 2$. Considérons le produit $\tilde{\mathbf{X}}\mathbf{v}$, qui résulte en un vecteur de taille $N\times 1$. Le i-ème élément du

produit $\tilde{\mathbf{X}}\mathbf{v}$ est :

$$\begin{aligned}(\tilde{\mathbf{X}}\mathbf{v})_i &= \sum_{j=1}^{p} \tilde{X}_{ij} v_j \\ &= \sum_{j \in J} \tilde{X}_{ij} \cdot 1 + \sum_{j \notin J} \tilde{X}_{ij} \cdot 0 \\ &= \sum_{j \in J} \tilde{X}_{ij}.\end{aligned}$$

Or, nous avons précédemment établi que $\sum_{j \in J} \tilde{X}_{ij} = 0 \ \forall \ i$. Par conséquent, $\tilde{\mathbf{X}}\mathbf{v} = \mathbf{0}_N$, où $\mathbf{0}_N$ est le vecteur nul dans $\mathbb{R}^N$.

Il suffit maintenant de démontrer que $\hat{\mathbf{S}}\mathbf{v} = \mathbf{0}_p$. Pour ce faire, calculons le produit $\hat{\mathbf{S}}\mathbf{v}$:

$$\begin{aligned}\hat{\mathbf{S}}\mathbf{v} &= \left(\frac{1}{N-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)\mathbf{v} \\ &= \frac{1}{N-1}\tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\mathbf{v}) \quad \text{(par associativité de la multiplication matricielle)} \\ &= \frac{1}{N-1}\tilde{\mathbf{X}}^T(\mathbf{0}_N) \quad \text{(en utilisant le résultat de précédent)} \\ &= \frac{1}{N-1}\mathbf{0}_p \\ &= \mathbf{0}_p\end{aligned}$$

Nous avons trouvé un vecteur non nul $\mathbf{v} \in \mathbb{R}^p$ tel que $\hat{\mathbf{S}}\mathbf{v} = \mathbf{0}_p$. Par définition, une matrice carrée A est singulière si et seulement s'il existe un vecteur non nul $\mathbf{x}$ tel que $A\mathbf{x} = \mathbf{0}$. Puisque nous avons démontré l'existence d'un tel vecteur pour la matrice de covariance empirique $\hat{\mathbf{S}}$, nous concluons que $\hat{\mathbf{S}}$ est singulière.

ANNEXE C DÉMONSTRATION DU RESPECT DE LA DÉFINITION D'UN ESPACE MÉTRIQUE POUR $(\mathcal{C}_i, L2)$

Un espace métrique est un ensemble muni d'une fonction de distance qui quantifie l'écart entre ses éléments. Notons $(\mathcal{C}_i, L2)$, un espace métrique pour le cluster $\mathcal{C}_i$ avec $L2$ la distance de type euclidienne, celle utilisée dans l'algorithme de résolution du problème k-centre. Pour que $(\mathcal{C}_i, L2)$ respecte la définition formelle d'un espace métrique, les trois propriétés suivantes doivent être respectées :

1. Symétrie : $\forall\ \mathbf{z}_i, \mathbf{z}_j \in \mathcal{C}_i,\ d(\mathbf{z}_i, \mathbf{z}_j) = d(\mathbf{z}_j, \mathbf{z}_i)$
2. Séparation : $\forall\ \mathbf{z}_i, \mathbf{z}_j \in \mathcal{C}_i,\ d(\mathbf{z}_i, \mathbf{z}_j) = 0 \iff \mathbf{z}_i = \mathbf{z}_j$
3. Inégalité triangulaire : $\forall\ \mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k \in \mathcal{C}_i,\ d(\mathbf{z}_i, \mathbf{z}_j) \leq d(\mathbf{z}_i, \mathbf{z}_k) + d(\mathbf{z}_k, \mathbf{z}_j)$

Respect de la symétrie

Considérons $\mathbf{z}_i = (z_i^{(1)}, \ldots, z_i^{(d_{latent})})$ et $\mathbf{z}_j = (z_j^{(1)}, \ldots, z_j^{(d_{latent})})$ deux vecteurs $\in \mathcal{C}_i$ et définis dans l'espace $\mathbb{R}^{d_{latent}}$ résultant de la réduction de dimension par auto-encodeur. La distance euclidienne entre $\mathbf{z}_i$ et $\mathbf{z}_j$ est définie par :

$$L2(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\sum_{k=1}^{d_{latent}} (z_i^{(k)} - z_j^{(k)})^2}. \tag{C.1}$$

Pour montrer la symétrie, il faut prouver que $L2(\mathbf{z}_i, \mathbf{z}_j) = L2(\mathbf{z}_j, \mathbf{z}_i)$. On sait d'abord que :

$$(z_i^{(k)} - z_j^{(k)})^2 = (-(z_j^{(k)} - z_i^{(k)}))^2 = (z_j^{(k)} - z_i^{(k)})^2\ \forall\ k \in \{1, 2, ..., d_{latent}\} \tag{C.2}$$

Ainsi, on peut écrire :

$$\sum_{k=1}^{d_{latent}} (z_i^{(k)} - z_j^{(k)})^2 = \sum_{k=1}^{d_{latent}} (z_j^{(k)} - z_i^{(k)})^2. \tag{C.3}$$

En appliquant la racine carrée des deux côtés de l'équation C.3, on obtient :

$$\sqrt{\sum_{k=1}^{d_{latent}} (z_i^{(k)} - z_j^{(k)})^2} = \sqrt{\sum_{k=1}^{d_{latent}} (z_j^{(k)} - z_i^{(k)})^2} \tag{C.4}$$

Or, il a été établi par l'équation C.1 que : $L2(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\sum_{k=1}^{d_{latent}} (z_i^{(k)} - z_j^{(k)})^2}$, d'où :

$$L2(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\sum_{k=1}^{d_{latent}} (z_i^{(k)} - z_j^{(k)})^2} = \sqrt{\sum_{k=1}^{d_{latent}} (z_j^{(k)} - z_i^{(k)})^2}, \tag{C.5}$$

ce qui montre que $L2(\mathbf{z}_i, \mathbf{z}_j) = L2(\mathbf{z}_j, \mathbf{z}_i)$. La propriété de symétrie est donc satisfaite.

Respect de la séparation

Pour établir la condition $L2(\mathbf{z}_i, \mathbf{z}_j) = 0 \iff \mathbf{z}_i = \mathbf{z}_j$, on utilise le fait que la somme des carrés de réels est nulle uniquement lorsque chacun des termes qui la composent est lui-même nul.

Supposons d'abord que $L2(\mathbf{z}_i, \mathbf{z}_j) = 0$, qui implique :

$$\sqrt{\sum_{k=1}^{d_{\text{latent}}} (z_i^{(k)} - z_j^{(k)})^2} \;=\; 0 \quad \Longrightarrow \quad \sum_{k=1}^{d_{\text{latent}}} (z_i^{(k)} - z_j^{(k)})^2 \;=\; 0. \tag{C.6}$$

Puisque chaque terme $\left(z_i^{(k)} - z_j^{(k)}\right)^2$ est non négatif, la somme ne peut être nulle que si chaque terme l'est également. On a donc :

$$(z_i^{(k)} - z_j^{(k)})^2 \;=\; 0 \quad \Longrightarrow \quad z_i^{(k)} = z_j^{(k)} \;\forall\; k. \tag{C.7}$$

Autrement dit, $\mathbf{z}_i$ et $\mathbf{z}_j$ coïncident exactement coordonnée par coordonnée, ce qui prouve $\mathbf{z}_i = \mathbf{z}_j$.

La réciproque peut se voir immédiatement : si $\mathbf{z}_i = \mathbf{z}_j$, alors $\left(z_i^{(k)} - z_j^{(k)}\right) = 0 \;\forall\; k$. Par conséquent :

$$\sum_{k=1}^{d_{\text{latent}}} (z_i^{(k)} - z_j^{(k)})^2 = 0 \;\;\Longrightarrow\;\; L2(\mathbf{z}_i, \mathbf{z}_j) = 0. \tag{C.8}$$

On conclut ainsi que $L2(\mathbf{z}_i, \mathbf{z}_j) = 0 \;\;\Longleftrightarrow\;\; \mathbf{z}_i = \mathbf{z}_j$, ce qui montre le respect de la propriété de séparation.

Respect de l'inégalité triangulaire

Considérons la norme euclidienne, qui vérifie la propriété de Minkowski :

$$\|\mathbf{u} + \mathbf{v}\|_p \leq \|\mathbf{u}\|_p + \|\mathbf{v}\|_p, \tag{C.9}$$

avec $p = 2$. Considérons 3 vecteurs distincts $\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k \in \mathcal{C}_i$ définis dans $\mathbb{R}^{d_{\text{latent}}}$ et posons : $\mathbf{u} = \mathbf{z}_i - \mathbf{z}_k$ et $\mathbf{v} = \mathbf{z}_k - \mathbf{z}_j$, où $\mathbf{u}$ et $\mathbf{v}$ sont les vecteurs dans l'équation C.9. On remarque

facilement que :

$$\mathbf{z}_i - \mathbf{z}_j = \mathbf{u} + \mathbf{v}. \tag{C.10}$$

En substituant ce résultat dans l'équation C.9, nous obtenons :

$$\|\mathbf{z}_i - \mathbf{z}_j\|_2 \leq \|\mathbf{z}_i - \mathbf{z}_k\|_2 + \|\mathbf{z}_k - \mathbf{z}_j\|_2. \tag{C.11}$$

En notation de distance, on peut réécrire :

$$(L2(\mathbf{z}_i, \mathbf{z}_j) \leq L2(\mathbf{z}_i, \mathbf{z}_k) + L2(\mathbf{z}_k, \mathbf{z}_j), \tag{C.12}$$

ce qui prouve que l'inégalité triangulaire est respectée pour la distance euclidienne et les points d'un cluster $\mathcal{C}_i$.

Les 3 démonstrations ci-haut certifient que les clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k$ et les ensembles contenant les distances euclidiennes entre leurs points, notés $(\mathcal{C}_1, L2), (\mathcal{C}_2, L2), ..., (\mathcal{C}_k, L2)$, constituent des espaces métriques.

ANNEXE D CORRÉLATION DE SPEARMAN INTRA-GROUPE

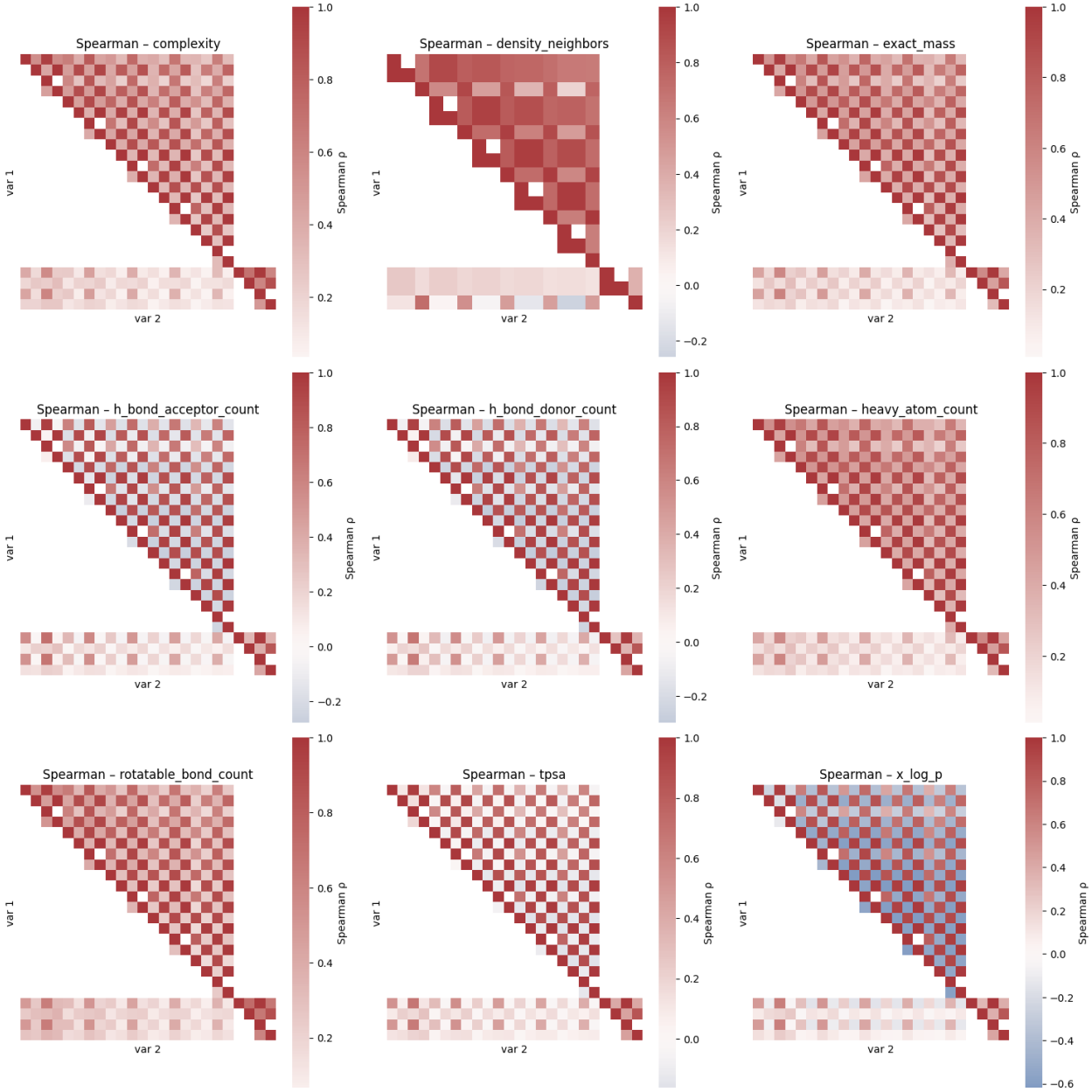

FIGURE D.1 Corrélation de Spearman intra-groupe.

ANNEXE E CONFIGURATION DES HYPERPARAMÈTRES DES MODÈLES DE CLASSIFICATION

- **Forêt aléatoire**
  - critère : ['gini', 'entropy']
  - max_variables : ['sqrt', 'log2']
  - n_arbres : [50, 100, 200, 300]
  - ccp_alpha : [10, 5, 1, 0.01, 0.05, 0.001]
- **Régression logistique**
  - C : [0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.05, 0.1]
  - pénalité : ['l1', 'l2']
- **XGBoost**
  - max_profondeur : [None, 3, 5, 7]
  - n_arbres : [100, 200, 300]
  - taux_apprentissage : [0.1, 0.01, 0.001]
  - alpha (régularisation) : [5, 1, 0.1]
- **RNA**
  - dimension_couches_cachées : [(200,), (150,), (50, 50)]
  - activation : [leakyrelu, relu, sigmoid, tanh]
  - optimiseur : ['adam']
  - alpha : [0.5, 0.1, 0.01]
  - taille_lot : [16, 32, 64]
  - taux_apprentissage_type : ['adaptive']
  - taux_apprentissage : [0.001, 0.01, 0.1, 0.5]

ANNEXE F TABLEAUX ET FIGURES DE RÉSULTATS ADDITIONNELS

TABLEAU F.1 Comparaison de l'AUC-PR pour différentes méthodes de sélection de variables en entraînement, validation croisée (VC) et test, scénario de priorisation de l'interprétabilité.

Sélection de variables	**Modèle**	**Entraînement PR-AUC**	**VC PR-AUC**	**Test PR-AUC**
Méthode originale	Régression Logistique	0.8468 (0.0030)	0.8515 (0.0037)	0.8508 (0.0035)
	RNA	0.8639 (0.0031)	0.8575 (0.0033)	0.8661 (0.0039)
	XGBoost	0.8800 (0.0032)	0.8709 (0.0035)	0.8712 (0.0036)
	Forêt aléatoire	0.8587 (0.0034)	0.8468 (0.0037)	0.8529 (0.0035)
	LDA	0.8168 (0.0031)	0.8253 (0.0036)	0.8236 (0.0037)
mRmR - anova & corrélation	Régression Logistique	0.7842 (0.0030)	0.7884 (0.0037)	0.7902 (0.0035)
	RNA	0.8024 (0.0031)	0.8027 (0.0033)	0.8014 (0.0039)
	XGBoost	0.7758 (0.0032)	0.7790 (0.0035)	0.7854 (0.0036)
	Forêt aléatoire	0.7863 (0.0034)	0.7882 (0.0037)	0.7866 (0.0035)
	LDA	0.7596 (0.0031)	0.7575 (0.0036)	0.7554 (0.0037)
mRmR - information mutuelle	Régression Logistique	0.7784 (0.0030)	0.7922 (0.0037)	0.7797 (0.0035)
	RNA	0.8004 (0.0031)	0.7896 (0.0033)	0.7960 (0.0039)
	XGBoost	0.7802 (0.0032)	0.7731 (0.0035)	0.7791 (0.0036)
	Forêt aléatoire	0.7881 (0.0034)	0.7812 (0.0037)	0.7809 (0.0035)
	LDA	0.7541 (0.0031)	0.7599 (0.0036)	0.7595 (0.0037)
Corrélation Spearman	Régression Logistique	0.7344 (0.0030)	0.7413 (0.0037)	0.7407 (0.0035)
	RNA	0.7655 (0.0031)	0.7673 (0.0033)	0.7598 (0.0039)
	XGBoost	0.7783 (0.0032)	0.7878 (0.0035)	0.7824 (0.0036)
	Forêt aléatoire	0.7709 (0.0034)	0.7685 (0.0037)	0.7662 (0.0035)
	LDA	0.7595 (0.0031)	0.7570 (0.0036)	0.7554 (0.0037)
Forêt aléatoire	Régression Logistique	0.8107 (0.0030)	0.8170 (0.0037)	0.8105 (0.0035)
	RNA	0.8394 (0.0031)	0.8474 (0.0033)	0.8397 (0.0039)
	XGBoost	0.8019 (0.0032)	0.8212 (0.0035)	0.8066 (0.0036)
	Forêt aléatoire	0.8224 (0.0034)	0.8224 (0.0037)	0.8181 (0.0035)
	LDA	0.7855 (0.0031)	0.7893 (0.0036)	0.7965 (0.0037)
Mann Whitney-U	Régression Logistique	0.7697 (0.0030)	0.7643 (0.0037)	0.7586 (0.0035)
	RNA	0.7971 (0.0031)	0.7996 (0.0033)	0.7906 (0.0039)
	XGBoost	0.7555 (0.0032)	0.7648 (0.0035)	0.7611 (0.0036)
	Forêt aléatoire	0.7672 (0.0034)	0.7612 (0.0037)	0.7644 (0.0035)
	LDA	0.7348 (0.0031)	0.7289 (0.0036)	0.7400 (0.0037)

TABLEAU F.2 Comparaison des différentes métriques en entraînement, validation croisée (VC) et test.

Métrique	**Modèle**	**Entraînement**	**VC**	**Test**
PR-AUC	Régression Logistique	0.8805 (0.0032)	0.8754 (0.0034)	0.8821 (0.0031)
	RNA	0.8851 (0.0029)	0.8993 (0.0031)	0.8926 (0.0033)
	XGBoost	0.8907 (0.0033)	0.8895 (0.0035)	0.9015 (0.0039)
	Forêt aléatoire	0.8723 (0.0035)	0.8702 (0.0032)	0.8689 (0.0034)
	LDA	0.8427 (0.0040)	0.8395 (0.0038)	0.8357 (0.0041)
ROC-AUC	Régression Logistique	0.8794 (0.0036)	0.8687 (0.0037)	0.8819 (0.0035)
	RNA	0.8752 (0.0034)	0.8904 (0.0032)	0.8945 (0.0035)
	XGBoost	0.8281 (0.0032)	0.8311 (0.0034)	0.8243 (0.0038)
	Forêt aléatoire	0.8607 (0.0033)	0.8632 (0.0031)	0.8632 (0.0034)
	LDA	0.8313 (0.0039)	0.8333 (0.0037)	0.8228 (0.0040)
Score F1	Régression Logistique	0.8803 (0.0033)	0.8568 (0.0035)	0.8684 (0.0036)
	RNA	0.8704 (0.0037)	0.8824 (0.0036)	0.8984 (0.0038)
	XGBoost	0.8281 (0.0032)	0.8311 (0.0034)	0.8243 (0.0038)
	Forêt aléatoire	0.8576 (0.0032)	0.8620 (0.0034)	0.8565 (0.0033)
	LDA	0.8322 (0.0041)	0.8180 (0.0040)	0.8259 (0.0042)
Exactitude	Régression Logistique	0.8698 (0.0035)	0.8589 (0.0033)	0.8532 (0.0036)
	RNA	0.8605 (0.0034)	0.8842 (0.0035)	0.8915 (0.0037)
	XGBoost	0.8207 (0.0034)	0.8240 (0.0036)	0.8215 (0.0040)
	Forêt aléatoire	0.8415 (0.0032)	0.8603 (0.0036)	0.8471 (0.0034)
	LDA	0.8310 (0.0038)	0.8117 (0.0039)	0.8254 (0.0040)
Précision	Régression Logistique	0.8668 (0.0037)	0.8598 (0.0034)	0.8475 (0.0038)
	RNA	0.8514 (0.0035)	0.8823 (0.0037)	0.8885 (0.0036)
	XGBoost	0.7783 (0.0036)	0.7878 (0.0038)	0.7824 (0.0042)
	Forêt aléatoire	0.8400 (0.0036)	0.8598 (0.0039)	0.8542 (0.0038)
	LDA	0.8242 (0.0042)	0.8037 (0.0041)	0.8328 (0.0043)
Rappel	Régression Logistique	0.8661 (0.0032)	0.8561 (0.0034)	0.8371 (0.0035)
	RNA	0.8466 (0.0030)	0.8871 (0.0032)	0.8764 (0.0036)
	XGBoost	0.8654 (0.0031)	0.8628 (0.0033)	0.8595 (0.0037)
	Forêt aléatoire	0.8399 (0.0033)	0.8541 (0.0032)	0.8427 (0.0034)
	LDA	0.8124 (0.0040)	0.7973 (0.0038)	0.8358 (0.0043)

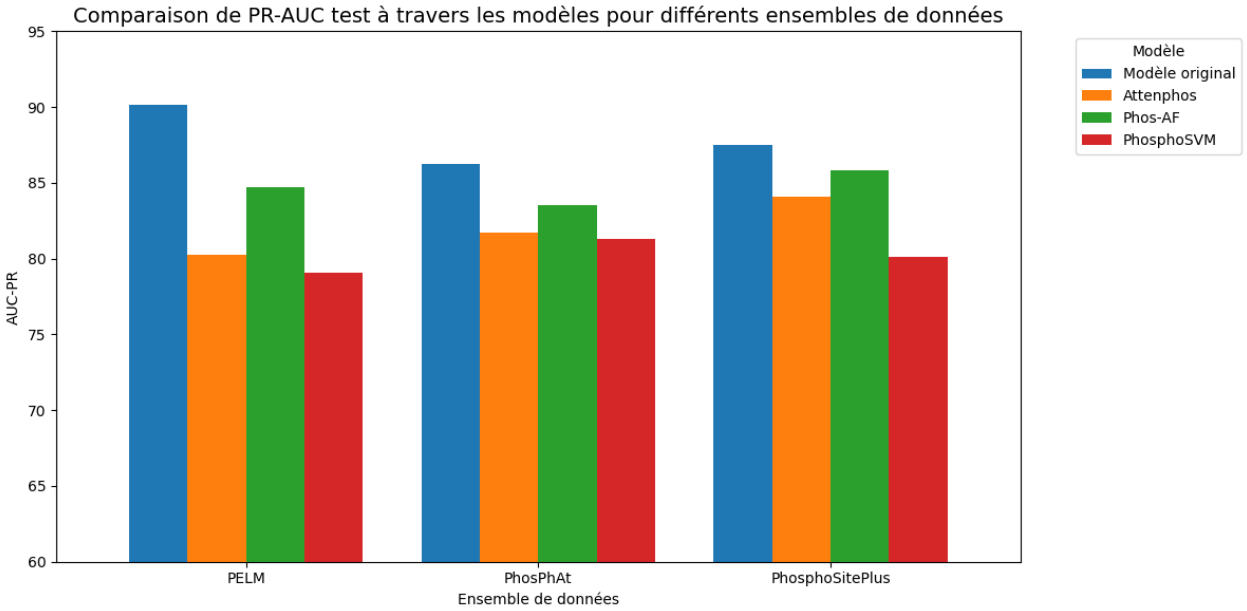

FIGURE F.1 Comparaison de PR-AUC test à travers différents modèles.

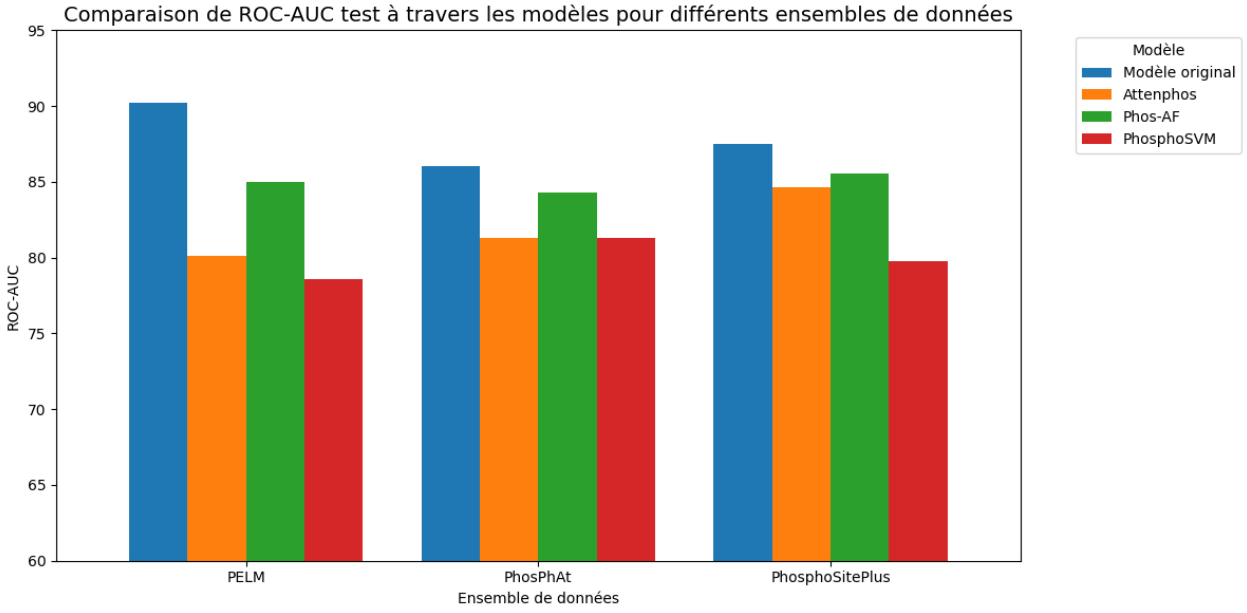

FIGURE F.2 Comparaison de ROC-AUC test à travers différents modèles.

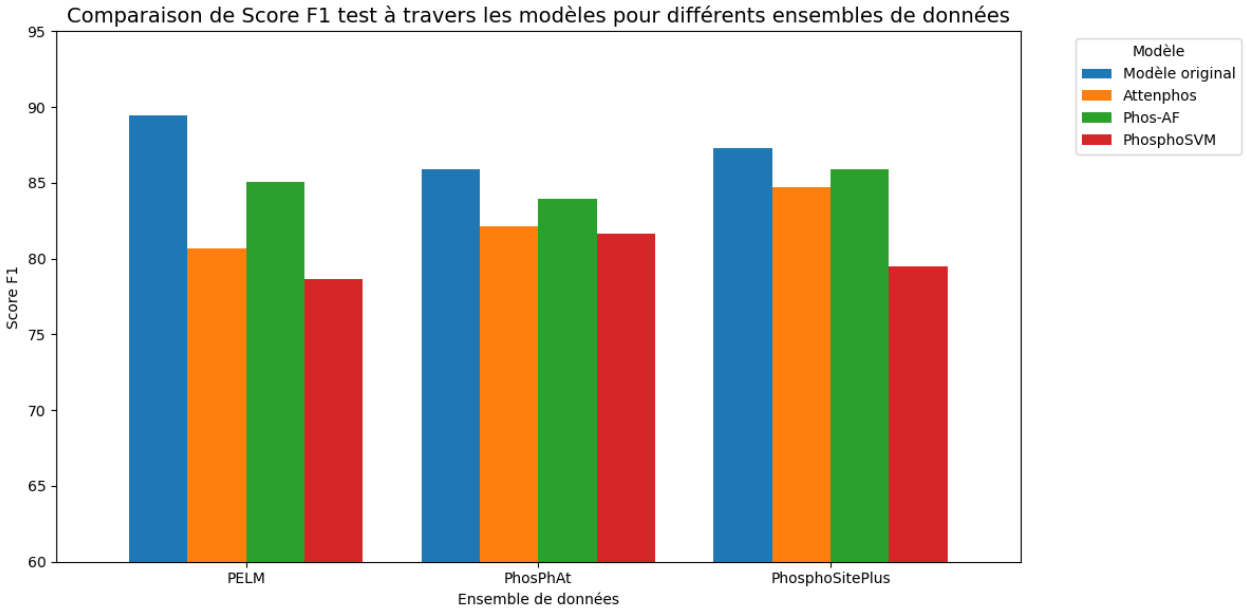

FIGURE F.3 Comparaison du score F-1 test à travers différents modèles.

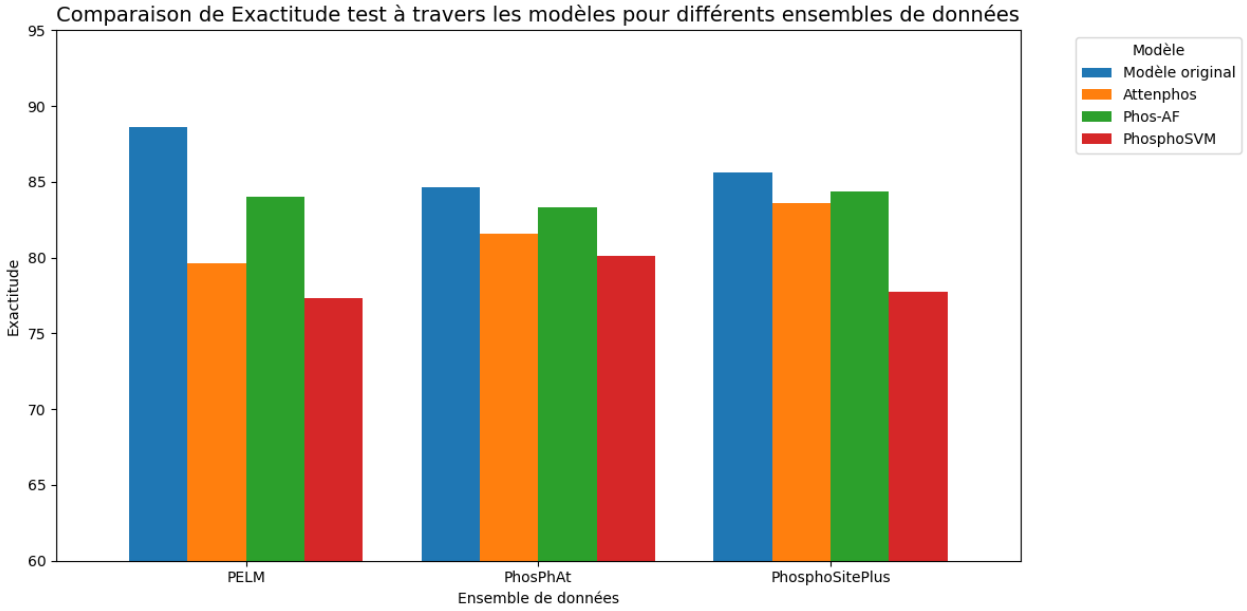

FIGURE F.4 Comparaison de l'exactitude test à travers différents modèles.

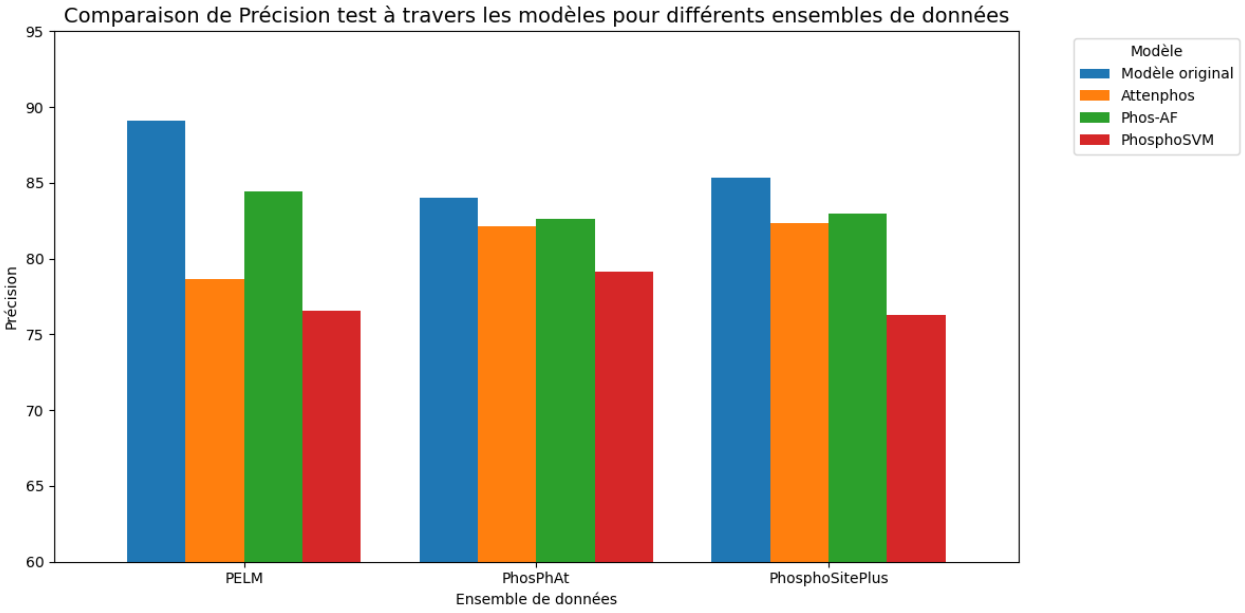

FIGURE F.5 Comparaison de la précision test à travers différents modèles.

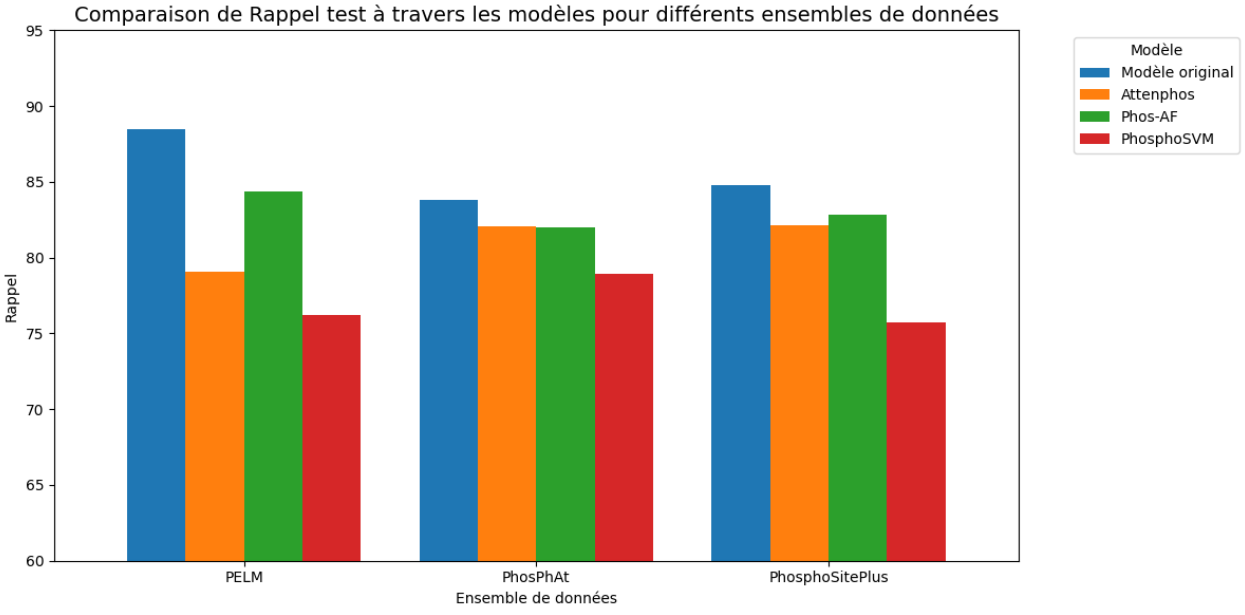

FIGURE F.6 Comparaison du rappel test à travers différents modèles.

ANNEXE G PROBABILITÉS DE PHOSPHORYLATION DES 42 RÉSIDUS CIBLÉS DE LA PROTÉINE TAU

TABLEAU G.1 Intervalles de confiance à 95% de la probabilité de phosphorylation pour tous les résidus de la protéine tau ciblés, ayant un potentiel élevé ou moyen.

Type d'acide aminé	**Position**	**Intervalle de confiance**	**Potentiel biomarqueur**
Thréonine	52	[0,9034 ; 0,9565]	Potentiel élevé
Sérine	137	[0,8804 ; 0,9341]	
Sérine	56	[0,8660 ; 0,9071]	
Thréonine	361	[0,8496 ; 0,8955]	
Thréonine	123	[0,8435 ; 0,8785]	
Thréonine	319	[0,7627 ; 0,8127]	Potentiel modéré
Sérine	113	[0,7410 ; 0,7910]	
Sérine	184	[0,6537 ; 0,7037]	
Thréonine	30	[0,6533 ; 0,7033]	
Thréonine	17	[0,6497 ; 0,6997]	
Thréonine	63	[0,6348 ; 0,6848]	
Tyrosine	310	[0,6302 ; 0,6802]	
Thréonine	149	[0,6055 ; 0,6555]	
Thréonine	386	[0,5966 ; 0,6466]	
Thréonine	245	[0,5283 ; 0,5850]	
Thréonine	39	[0,5282 ; 0,5708]	

TABLEAU G.2 Intervalles de confiance à 95% de la probabilité de phosphorylation pour les résidus de la protéine tau ciblés, ayant un faible potentiel de servir de biomarqueur.

Type d'acide aminé	**Position**	**Intervalle de confiance**	**Potentiel biomarqueur**
Thréonine	220	[0,3977 ; 0,4398]	Potentiel faible
Sérine	412	[0,3837 ; 0,4369]	
Thréonine	102	[0,3536 ; 0,3926]	
Thréonine	50	[0,3470 ; 0,3946]	
Thréonine	377	[0,3328 ; 0,3782]	
Thréonine	76	[0,3326 ; 0,3800]	
Sérine	129	[0,3020 ; 0,3351]	
Thréonine	263	[0,2999 ; 0,3468]	
Thréonine	101	[0,2900 ; 0,3304]	
Sérine	238	[0,2695 ; 0,3086]	
Sérine	64	[0,2398 ; 0,2838]	
Sérine	131	[0,1947 ; 0,2373]	
Sérine	208	[0,1940 ; 0,2297]	
Sérine	210	[0,1915 ; 0,2457]	
Sérine	68	[0,1835 ; 0,2148]	
Thréonine	414	[0,1729 ; 0,2145]	
Sérine	413	[0,1623 ; 0,2033]	
Thréonine	102	[0,1616 ; 0,2148]	
Thréonine	95	[0,1486 ; 0,1987]	
Sérine	241	[0,1473 ; 0,1879]	
Sérine	289	[0,1186 ; 0,1579]	
Sérine	320	[0,0923 ; 0,1402]	
Sérine	341	[0,0875 ; 0,1320]	
Sérine	316	[0,0819 ; 0,1231]	
Sérine	435	[0,0681 ; 0,1092]	
Sérine	433	[0,0571 ; 0,1013]	