Rapport de stage – Master 1 BI-IPFB

Année universitaire 2024–2025

Nom du candidat

Karim Bouchaara

Titre du mémoire

Reproduction partielle de la pipeline RFpeptides pour le design de peptides cycliques

(d'après Rettie et al., 2024 – "Accurate de novo design of high-affinity protein binding macrocycles using deep learning")

Laboratoire d'accueil

UMR BFA (CNRS 8251, Université Paris Cité)

Responsable scientifique

Dirk Stratmann

Adresse du laboratoire

4 Rue Marie-Andrée Lagroua Weill-Hallé
75013 Paris, France

Contents

1 Introduction

Les peptides cycliques suscitent un intérêt croissant en biotechnologie en raison de leur stabilité structurale, de leur résistance aux dégradations enzymatiques et de leur aptitude à adopter des conformations compactes, favorables aux interactions avec des cibles biologiques. Leur géométrie fermée leur confère une rigidité conformationnelle accrue, réduisant l'entropie lors de la reconnaissance moléculaire et augmentant l'affinité. Ces propriétés en font des candidats prometteurs pour le développement de nouvelles entités thérapeutiques ciblant des interfaces protéiques complexes, peu accessibles aux petites molécules ou aux anticorps classiques[1].

La conception de novo de peptides cycliques reste cependant une tâche difficile. Les approches expérimentales comme le criblage à haut débit ou le phage display permettent d'identifier des ligands fonctionnels, mais leur couverture de l'espace structural reste limitée. Du côté computationnel, la majorité des outils existants sont conçus pour modéliser des structures linéaires, avec des extrémités N-terminales et C-terminales explicites. Cette hypothèse n'est pas adaptée aux peptides cycliques, dans lesquels ces extrémités sont liées, rendant nécessaire une représentation alternative.Pour répondre à cette limite, plusieurs outils ont été récemment proposés. AfCycDesign [2] permet à la fois la prédiction de structure, le redesign de séquence sur un squelette donné, et le de novo binder design par hallucination conformationnelle et séquentielle. Contrairement à une approche nécessitant une séquence initiale, AfCycDesign peut générer directement des peptides capables de se lier à des interfaces protéiques à partir de contraintes structurales ou de motifs fonctionnels.CyclicBoltz1 [3] utilise des réseaux de neurones géométriques pour prédire des structures de peptides macrocycliques, y compris ceux contenant des acides aminés non naturels.

Dans ce contexte, le pipeline RFpeptides, proposé par Rettie et al. [4], offre une solution complète pour la génération, le design et la validation de peptides cycliques. Il s'appuie sur plusieurs modules issus de l'intelligence artificielle, initialement développés par le laboratoire de David Baker (Institute for Protein Design, University of Washington). Ce pipeline comporte trois étapes principales : (i) la génération de la structure 3D du squelette peptidique via RFdiffusion (Wang et al. (2025), un modèle de diffusion guidée par structure ; (ii) le design de la séquence compatible à l'aide de ProteinMPNN (Lin et al. (2023)), un réseau de neurones dédié ; et (iii) la validation conformationnelle avec RosettaFold2 (Rettie, S. A. et al. (2024)) ou AfCycDesign. L'originalité de RFpeptides réside dans l'introduction d'un encodage cyclique relatif, c'est-à-dire une modification du codage positionnel dans le réseau neuronal, permettant au modèle de considérer le peptide comme un cycle fermé, sans extrémité N-ter ou C-ter. Cela évite les discontinuités dans la chaîne peptidique et permet la génération de structures cycliques géométriquement réalistes. Le pipeline de RFpeptides est résumé dans la Figure 1, qui présente les différentes étapes, depuis l'entrée d'un motif peptidique jusqu'à l'analyse de la structure générée. Lors de ce stage nous avons mis en place sur le cluster local de RPBS les trois premières étapes (RFdiffusion, ProteinMPNN et RosettaFold2), qui ont été ensuite enchaîné dans un pipeline automatique et ainsi testé individuellement

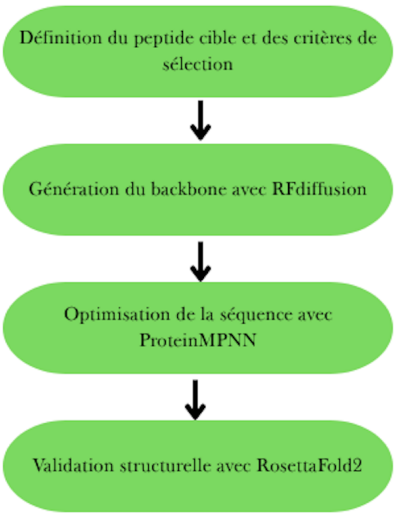


Figure 1: Schéma du pipeline RFpeptides pour la conception assistée par IA de peptides cycliques à partir d'un motif cibleLe pipeline RFpeptides repose sur l'utilisation successive de RFdiffusion (génération structurale), ProteinMPNN (design de séquence), RosettaFold2

et conjointement.

1.1 Objectifs du stage

Le stage présenté dans ce rapport avait comme objectif de reproduire une partie des travaux de Rettie et al sur RFpepitdes. Il a été réalisé au sein de l'UMR BFA (CNRS 8251, Université Paris Cité), sous la direction de Dirk Stratmann, avec pour objectif d'implémenter, tester et documenter le pipeline RFpeptides. Il ne s'agissait pas de développer un nouvel outil, mais d'évaluer la robustesse du protocole sur des cas concrets de design peptidique, en particulier dans le cadre de la génération de peptides cycliques à partir de fragments ou de motifs structuraux. Tester ce pipeline sur des cas représentatifs extraits de structures expérimentales connues (PDB) permet d'évaluer sa généricité, sa reproductibilité et ses limites dans le contexte du design computationnel de peptides cycliques. Le rapport présente successivement les outils employés, la mise en place technique, les résultats obtenus pour chacun des scénarios (scaffolding autour d'un motif, remodelage de boucle, ou design contraint par une interface), ainsi qu'une discussion critique sur les apports et limites du pipeline.

2 Matériels et Méthodes

Cette section décrit, dans un premier temps, les outils utilisés et leur rôle respectif dans le pipeline, puis détaille le protocole global mis en place, les cas testés, et enfin les conditions de calcul employées pour l'exécution des différentes étapes.

Outil	Rôle principal	Entrée	Sortie	Principe
RFdiffusion (Wang et al. (2025))	Génération de backbone peptidique	Motif cible, contraintes	Structure 3D du squelette (poly-Gly)	Méthode de diffusion structure-guidée
ProteinMPNN (Lin et al. (2023))	Design de séquence	Structure 3D du squelette généré (ou du complexe peptide-cible)	Séquence peptidique	Réseau de neurones feed-forward entraîné sur des structures natives
RosettaFold2 (Rettie, S. A. et al. (2024))	Validation conformationnelle	Séquence peptidique et structure cible	Structure 3D prédite	Transformeur multi-chaînes, basé sur les principes d'AlphaFold2, tel qu'implémenté dans la version modifiée de RosettaFold2 présentée par Rettie et al. (2024)

Table 1: Outils utilisés dans le pipeline RFpeptides pour la génération et l'évaluation de peptides cycliques. Chaque outil est intégré à une étape spécifique du protocole, avec des entrées et sorties précises, permettant une automatisation du processus de conception assistée par IA.

2.1 Les trois premières étapes de RFpepitdes

Le pipeline RFpeptides repose sur une suite d'outils développés pour automatiser la génération de peptides cycliques. Chaque module du pipeline remplit une fonction spécifique, allant de la génération de la structure peptidique à la validation conformationnelle. L'ensemble a été exécuté étape par étape, en ligne de commande, dans un environnement Linux. Le tableau 1 résume les rôles, entrées et sorties principales de chaque outil. Chaque outil correspond à une étape du pipeline (voir Figure 1). RFdiffusion permet de générer une conformation 3D plausible d'un peptide fermé à partir d'un motif ou d'une contrainte spatiale. ProteinMPNN est ensuite utilisé pour déterminer une séquence en acides aminés compatible avec cette structure. La séquence obtenue est ensuite validée par RosettaFold2, qui prédit sa capacité à adopter la conformation souhaitée.

2.2 Étape 1 – Génération du backbone avec RFdiffusion

La première étape du pipeline RFpeptides consiste à générer une structure 3D du squelette d'un peptide cyclique à partir d'un motif ou de contraintes structurales. Elle s'appuie sur RFdiffusion, un générateur de structures de novo basé sur un modèle de diffusion. Initialement conçu pour des chaînes protéiques linéaires avec extrémités libres, RFdiffusion a été modifié dans RFpeptides pour tenir compte du caractère fermé des peptides cycliques, grâce à l'introduction d'un encodage

cyclique relatif. Cet encodage cyclique est une adaptation du codage positionnel des réseaux de neurones, qui permet de représenter les résidus non plus selon une séquence linéaire, mais comme un cycle, sans début ni fin. Il supprime ainsi la dépendance aux extrémités, essentielle pour modéliser correctement les peptides cycliques. Cette version modifiée de RFdiffusion n'étant pas encore publique, j'ai utilisé la version standard du modèle. RFdiffusion fonctionne selon un principe de génération progressive. Il commence par une structure complètement bruitée, dans laquelle la position des atomes est aléatoire, puis il applique une série d'étapes de débruitage pour converger vers une structure plausible, guidée par les contraintes fournies. Ces contraintes peuvent correspondre à un fragment 3D fixé dans l'espace, à des informations de contact entre résidus, ou à des contraintes de forme plus globales. Pour se familiariser avec RFdiffusion, plusieurs stratégies de génération ont été testées durant le stage.

Dans la stratégie dite de "motif scaffolding", une portion structurale connue est utilisée comme point d'ancrage. Elle est fixée en entrée et reste inchangée tout au long de la génération, tandis que le reste du peptide ou protéine est construit autour. Cette approche est utile lorsqu'un motif fonctionnel (comme une boucle de liaison ou un épitope) doit être maintenu dans la structure finale. J'ai utilisé un fragment extrait de la structure 5TPN comme motif à maintenir. Celui-ci était encodé en tant que contig fixe, avec une position définie dans l'espace, autour duquel RFdiffusion devait générer le reste du peptide. Pour chaque run, j'ai généré entre 50 et 200 structures candidates, avec différents niveaux de bruit (seed aléatoire) afin de maximiser la diversité des backbones. Dans le cas du "loop remodeling", une boucle d'une protéine connue est retirée, et le modèle est utilisé pour générer une nouvelle connexion entr les segments restants, en respectant les contraintes géométriques de la structure globale. Pour le scénario de loop remodeling, j'ai préparé manuellement des fichiers PDB dans lesquels une boucle cible avait été supprimée. Les positions des extrémités restantes étaient alors utilisées comme point d'ancrage pour la génération d'un segment de remplacement, dans le but de reconnecter les chaînes Des premiers essais ont permis de générer des conformations compatibles avec l'interface, mais sans validation complète dans cette version du pipeline. Enfin, dans le scénario de "interface design", RFdiffusion est utilisé pour produire une conformation qui soit compatible avec une surface de liaison ciblée. Les entrées de RFdiffusion sont des fichiers structurés décrivant les contraintes (souvent appelés "contigs", c'est-à-dire des segments structuraux définis), ainsi que des paramètres comme la longueur souhaitée de la séquence, le nombre d'échantillons à produire, et le niveau de bruit initial. Le modèle génère en sortie un ensemble de structures tridimensionnelles poly-Gly au format PDB, représentant différentes conformations candidates sans les chaînes latérales. Ces structures sont ensuite choisies selon des critères simples tels que la compacité, ou l'absence de conflits stériques (clashs). Les structures jugées correctes sont ensuite transmises à l'étape suivante du pipeline pour la conception de la séquence.

2.3 Étape 2 – Design de séquence avec ProteinMPNN

La deuxième étape du pipeline RFpeptides vise à concevoir une séquence d'acides aminés susceptible de stabiliser la structure peptidique générée précédemment. Cette opération a été effectuée à l'aide de ProteinMPNN. L'outil prend en entrée un fichier de structure tridimensionnelle (au format PDB), dans lequel seul le squelette du peptide est présent — typiquement sous forme de poly-glycines. À partir de cette géométrie, ProteinMPNN propose une ou plusieurs séquences d'acides aminés susceptibles d'adopter cette conformation, sans requérir d'information sur la séquence naturelle Par défaut, ProteinMPNN effectue un design de séquence sur une structure donnée sans distinction explicite entre un peptide libre ou en complexe. Toutefois, si la structure complète d'un complexe peptide–protéine est fournie en entrée, le logiciel peut intégrer les contacts spatiaux avec la protéine partenaire, dans la mesure où ceux-ci affectent la géométrie locale autour du peptide. Il n'optimise cependant pas directement les interactions intermoléculaires.

Chaque fichier généré par ProteinMPNN contenait une séquence en format FASTA, ce qui est standard pour les fichiers de séquences, accompagnée d'informations sur la chaîne modifiée (nom, identifiant dans le PDB) et de plusieurs scores d'évaluation. Parmi ces scores figurent notamment : le score total (log-likelihood de la séquence générée), les probabilités individuelles par position, et parfois un score de perplexité. Ces scores reflètent à quel point la séquence est compatible statistiquement avec la structure fournie, selon la distribution interne du modèle. Afin d'évaluer la cohérence structure-séquence de manière plus physique, des analyses complémentaires ont été réalisées. ProteinMPNN ne prédit pas la structure 3D associée à une séquence, il est donc nécessaire de passer par un outil externe comme RosettaFold2 pour générer la structure correspondant à la séquence produite. Une fois cette prédiction effectuée, un RMSD peut être calculé entre la structure initiale (celle du squelette poly-Gly) et la structure prédite à partir de la séquence générée, afin d'évaluer la fidélité du repliement. Le RMSD est en général calculé uniquement sur les atomes du squelette (backbone), car les chaînes latérales changent selon la séquence.Dans certains cas, une énergie de repliement peut également être calculée à partir de la structure 3D obtenue.

2.4 Étape 3 – Validation conformationnelle avec RoseTTAFold2 (RF2)

L'étape suivante du pipeline consistait à vérifier si les séquences générées pouvaient se replier correctement selon la structure prévue. Pour cela, nous avons utilisé uniquement RoseTTAFold2 (RF2), car il peut comme AfCycDesign prédire les structures des peptides cycliques. RFpeptides utilise les deux en parallèle pour une meilleure filtration des séquences générees, mais pour la preuve de concept nous nous sommes concentrés sur RF2, sans installer ou tester AfCycDesign. Une option spécifique déjà incluse dans RF2 permet de forcer la cyclisation de la chaîne peptidique (-cyclize).Par la suite, RF2 a été intégré dans le pipeline complet. Les séquences produites par ProteinMPNN ont été automatiquement repliées par RF2.

Pour vérifier le bon fonctionnement de RF2 sur les peptides cycliques, une tâche importante du

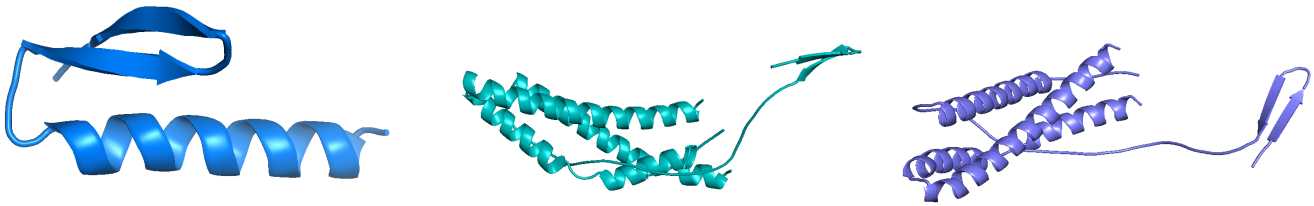
stage a été de reproduire les données de la figure S1 du preprint sur RFpeptides[4](voir annexe). Les auteurs ont testé RF2 sur 80 structures RMN de peptides cycliques, cyclisés N-ter / C-ter, contenant uniquement des acides aminés naturels et de taille inférieure à 40 acides aminés. Les structures peuvent également contenir un ou pluiseurs ponts disulfures. Les 80 codes PDB étaient disponible dans les SI du preprint sur RFpepitdes. Nous avons avec notre installation locale de RF2 calculé les structures de ces 80 peptides cycliques. Les structures générées ont ensuite été comparées aux structures natives expérimentales disponibles dans la base PDB, en utilisant l'indicateur RMSD (Root-Mean-Square Deviation) entre les deux squelettes (i.e. les chaînes latérales ne sont pas inclut dans le calcul du RMSD ici). Pour cela, un travail de traitement préalable a été nécessaire : les structures RMN, qui contiennent souvent plusieurs modèles dans un seul fichier, ont été séparées en fichiers individuels afin de permettre la comparaison.Des scripts python et bash ont été mis en place pour superposer chaque modèle prédictif à chacun des modèles RMN, et calculer le RMSD pour chaque alignement. Pour le calcul du RMSD proprement dit, nous nous sommes basés sur le script Rosetta XML fournit dans la partie SI du preprint de RFpeptides, qui s'utilise avec Rosetta classique de RosettaCommons. Pour chaque peptide, seule la meilleure correspondance (RMSD minimal) a été retenue, comme décrit dans la partie SI du preprint sur RFpeptides auquel nous nous référions. Ce processus a permis d'évaluer de manière systématique la qualité des structures générées.Une carte croisant les scores de confiance (pLDDT) et les valeurs de RMSD a été construite à partir des 80 structures. et les trois structures de la figure S1 (annexe) ont été analysé en détail.

3 Résultats et Discussion

3.1 Résultats de design avec RFdiffusion

Dans une première tâche, RFdiffusion a été utilisé pour insérer un motif structuré au sein d'un échafaudage protéique généré de novo (motif scaffolding). La figure 2a montre un exemple où l'algorithme a construit une structure cohérente autour du motif imposé. L'architecture globale, dominée ici par une hélice , s'adapte géométriquement au motif afin de le maintenir en place. Toutefois, la stabilité de cette hélice ne peut pas être garantie sans validation dynamique : une simulation de dynamique moléculaire serait nécessaire pour évaluer si elle reste bien ancrée à l'échafaudage dans des conditions réalistes. L'ensemble de la structure respecte par ailleurs la connectivité covalente du squelette, c'est-à-dire la continuité des liaisons peptidiques entre les résidus.

Dans une seconde tâche, RFdiffusion a été utilisé pour générer une structure continue entre deux segments protéiques en remodelant la boucle de liaison. La figure 2b montre une structure obtenue en imposant une conformation fixée sur un segment structuré. RFdiffusion a généré non seulement la boucle de liaison manquante, mais aussi des segments hélicoïdaux de part et d'autre, assurant la continuité de la chaîne et la stabilité globale de l'échafaudage. Dans une troisième tâche,



(a) (b) (c)

Figure 2: Résultats avec RFdiffusion: (a) Structure générée par RFdiffusion en mode motif scaffolding. Représentation 3D d'un échafaudage protéique généré par RFdiffusion à partir d'un motif de type feuillet . La structure complète a été reconstruite en maintenant la conformation locale du motif (b) Exemple de structure générée par RFdiffusion en mode « loop remodeling ». Une conformation locale est imposée sur un segment structuré (non distingué ici), et RFdiffusion génère une connectivité structurale complète, incluant une boucle allongée reliant plusieurs hélices . (c) Structure générée par RFdiffusion en mode binder deisgn . Le motif (à droite) a été extrait d'une interface réelle (ex. : 5TPN), et fixé comme contrainte spatiale. RFdiffusion a généré un échafaudage compatible autour de ce motif. La protéine partenaire n'est pas représentée ici, mais la structure cible est connue.

RFdiffusion a été employé pour construire un échafaudage protéique autour d'un segment structuré représentant une interface de liaison. La figure 2c illustre une structure dans laquelle un motif de feuillet , imposé comme contrainte initiale, est intégré à l'extrémité d'un repliement hélicoïdal généré de novo. La structure illustrée ici a été générée en fixant un motif d'interface extrait d'un complexe réel, mais sans inclure explicitement la protéine partenaire dans le modèle. L'absence de la cible dans la structure générée rend difficile l'évaluation directe de la stabilité de l'interaction. Cette approche permet d'envisager la conception de protéines interagissant avec des cibles définies, en maintenant la géométrie du site de liaison tout en générant un échafaudage stable autour. Dans une dernière étape préparatoire du pipeline de design, la structure 5TPN, correspondant à un complexe protéine–peptide a été utilisée pour extraire un motif conservé, situé à l'interface de liaison avec un partenaire protéique. Ce motif, visualisé en cyan sur la Figure 3, est constitué de plusieurs brins porteurs de résidus clés (en rouge) impliqués dans l'interaction. Ces résidus ont été sélectionnés pour être maintenus fixes durant le processus de génération afin de préserver le potentiel de liaison. La structure complète du complexe est représentée en gris pour situer le motif dans son contexte natif. Cette préparation permet d'alimenter RFdiffusion avec une contrainte locale à reproduire, tout en laissant l'algorithme proposer de nouvelles architectures compatibles avec l'interface cible. Cette stratégie technique correspond à un cas de motif scaffolding, puisque seul un fragment d'interface est fixé en entrée sans inclure la protéine cible.

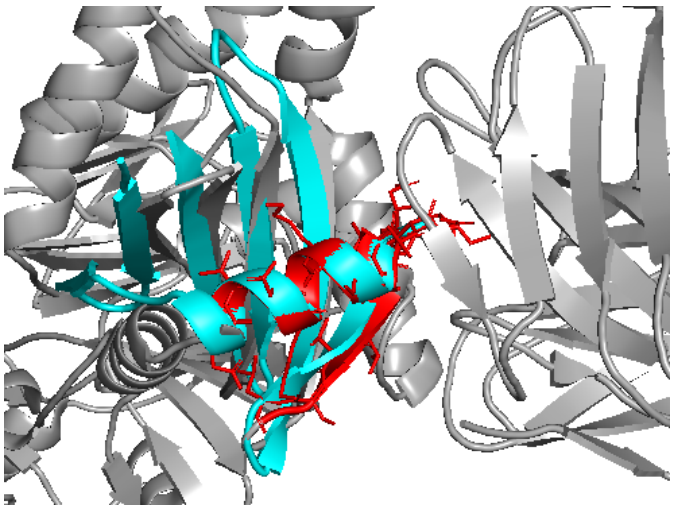


Figure 3: Motif extrait de 5TPN pour le design guidé par RFdiffusion. Le motif (en cyan) est extrait de la structure native 5TPN et utilisé comme contrainte structurale dans le design d'un nouveau binder.Les résidus impliqués dans l'interaction (en rouge) sont conservés durant la génération.

```
Query_10001   1   GGGGGGGGGGGEVNKIKSALLSTNKAVVSLGGGGGGGGGGGGGGGGGGGGG   54
Query_10002   1   KKIIFKELPEEEKKKLLEALKNKQNQIVVLSNGKKVLVSEAIINGKKVLVLKEL   54
Query_10003   1   KKYIYKELPKEELKKIEEALKNKQNQIVTLSDGRKYLVSKEIINGKEVYILKPL   54
Query_10004   1   KIVVFEELPEEELKKMEEALKNKQNQEVTLSDGKKYLVSKAIIKGKEVYVRKEL   54
```

Figure 4: Séquences conçues par ProteinMPNN à partir de la structure générée dans le cadre du motif scaffolding. La séquence de départ, composée essentiellement de glycines encadrant un motif cible, a été utilisée comme gabarit pour générer trois séquences plausibles à température de sampling T=0.1. Les séquences résultantes montrent des enrichissements cohérents en résidus polaires et chargés. Image crée par l'outil d'alignement multiple du site web NCBI Blast (en rouge résidus conservés).

3.2 Design de séquences avec ProteinMPNN

Dans un second temps, nous avons utilisé le modèle généré par RFdiffusion en mode motif scaffolding pour concevoir des séquences compatibles à l'aide de ProteinMPNN. La structure initiale, constituée majoritairement de glycines et de quelques résidus du motif d'intérêt, devait être convertie en une séquence plausible, capable d'adopter la conformation imposée. Trois séquences ont été générées à température de sampling 0.1, chacune associée à un score de compatibilité avec la structure. Elles présentent un enrichissement en acides aminés polaires ou chargés, typiques de structures solubles et stables. Bien que les différences se situent surtout dans les régions centrales, une certaine régularité d'occurrence est maintenue. La séquence de départ et les trois variantes générées sont présentées ci-dessous en format FASTA.

Pour tester la capacité de ProteinMPNN à générer de nouvelles séquences compatibles avec des motifs cycliques simples, nous avons utilisé comme point de départ la structure d'un peptide

cyclique disponible dans la PDB (code 3AVF), dont la séquence originale est DLKIDNLD. Dans la structure PDB ce peptide est lié à une protéine. ProteinMPNN a généré une nouvelle séquence en imposant une température de sampling très basse (T = 0.0001), ce qui limite fortement l'exploration aléatoire et pousse le modèle à proposer une séquence unique jugée optimale. La séquence produite, RNLPEDPA, diffère totalement de l'originale, ce que confirme un score de récupération de séquence (seq_recovery) nul. Il n'existe pas, dans ProteinMPNN, de mode natif permettant d'optimiser une séquence d'entrée sans la changer entièrement. Toutefois, il est possible de fixer certains résidus (via un masque booléen) afin d'en conserver une partie, ce qui permet un design partiellement contraint. Modifier la température de sampling permet également de générer des séquences plus proches de la distribution moyenne, mais cela ne garantit pas une similarité avec la séquence d'origine. Cela montre que ProteinMPNN peut concevoir des séquences plausibles qui respectent la forme de la structure cible, même lorsqu'elles sont complètement différentes de la séquence naturelle.

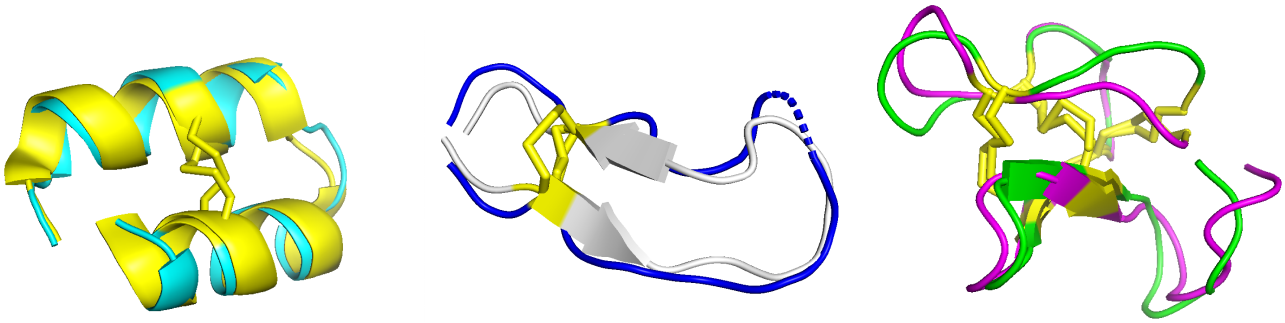
3.3 Résultats de prédiction avec RoseTTAFold2

Un premier ensemble d'analyses a porté sur la structure prédite par RoseTTAFold2 (RF2) pour le peptide cyclique 5KWZ, issu de la figure S1 du preprint de RFpeptides (voir annexe). Sur la figure 5a, la structure prédite (en jaune) est superposée à la structure expérimentale (en cyan; modèle RMN 1). On observe un bon recouvrement global des hélices , ainsi qu'une conservation du pont disulfure, représenté ici sous forme de bâtonnets. Cette cohérence structurale suggère que RF2 est capable de générer des repliements fiables, y compris dans le contexte de peptides cyclisés de petite taille (26 acides aminés ici). Le RMSD obtenu est de 0.754 Å sur le squelette, ce qui est très proche de la valeur reportée dans la figure S1 du preprint (0.8 Å). Cette faible différence confirme la reproductibilité des résultats et la précision du repliement généré dans ce cas. Une telle valeur reflète une quasi-identité structurale, ce qui est remarquable pour un modèle généré de novo en condition cyclique.

Le respect de la conformation native est particulièrement encourageant pour des applications de design de novo, où la précision de la topologie globale et des contraintes locales (comme ici les ponts disulfures) est essentielle à la stabilité et à la fonction de la molécule.

La figure 5b illustre la superposition entre la structure expérimentale du peptide 2NB5 (en blanc) et la structure prédite par RF2 en mode cyclisé (en jaune). Le pont disulfure, bien identifié dans la structure prédite, est représenté en bâtonnets jaunes. Cette liaison contribue fortement à la stabilité globale du peptide. Le RMSD de 1.796 Å reste inférieur à 2 Å (proche de 0,9 obtenu dans le preprint) , ce qui est acceptable, mais signale une variabilité locale. Cela peut s'expliquer par une flexibilité intrinsèque des boucles ou une moindre contrainte imposée dans ces régions lors de la prédiction.

On observe une très bonne conservation de la conformation centrale, notamment au niveau



(a) (b) (c)

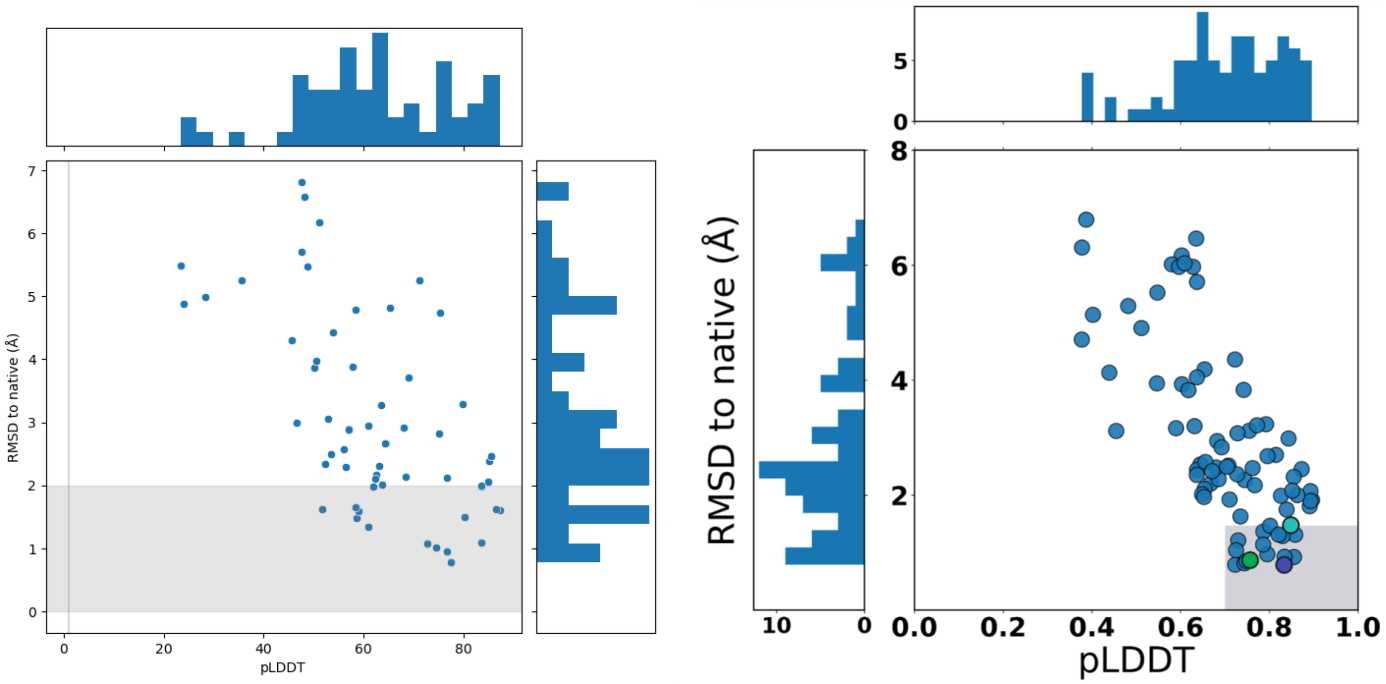
Figure 5: Prédiction des structures de peptides cycliques avec RF2. Les ponts disulfures sont représentés en bâtonnets. (a) Superposition entre la structure native du peptide 5KWZ (en cyan, quel modèle ?) et la structure prédite par RF2 en mode cyclisé (en jaune). (b) Superposition entre la structure native de 2NB5 (en blanc) et le modèle cyclisé généré par RF2 (en jaune). (c) Superposition de la structure native du peptide 2B38 (représentée en vert et magenta) avec une structure prédite cyclisée par RF2 (en jaune).

du feuillet , malgré des divergences plus marquées au niveau des extrémités. Cela témoigne de la capacité de RF2 à retrouver des motifs structuraux stabilisés dans des peptides cycliques, en particulier lorsqu'un pont disulfure est présent pour contraindre le repliement.

Enfin, la prédiction pour 2B38 (figure 5c) montre une topologie plus divergente. Bien que le pont disulfure soit correctement modélisé, et que les éléments secondaires principaux soient présents, les boucles et chaînes latérales adoptent des conformations distinctes de celles du modèle expérimental. Le RMSD de 1.553 Å traduit une divergence modérée, en accord avec l'impression visuelle : le cœur structuré est aligné, mais les régions périphériques s'écartent davantage. Cela met en évidence les limites de RF2 lorsqu'il s'agit de prédire des peptides très compacts avec des contraintes spécifiques :

Conclusion sur les trois structures. L'analyse des trois peptides met en évidence la capacité de RoseTTAFold2 à prédire avec précision la structure de certains peptides cycliques, comme 5KWZ, pour lequel le repliement est quasi parfait. D'autres cas, comme 2NB5 et surtout 2B38, montrent des divergences croissantes, en particulier dans les boucles ou régions flexibles. Ces variations reflètent les limites du modèle selon la complexité structurale ou la compacité du peptide. La fiabilité de RF2 reste donc variable et dépend du contexte, ce qui justifie l'usage de critères complémentaires pour filtrer les prédictions.

La figure 6 présente un nuage de points représentant les structures générées par RF2 pour le même ensemble de 80 peptides cycliques que dans la figure S1 (voir annexe). Chaque point correspond à une structure, positionnée selon son score pLDDT (indicateur de confiance interne retourné par RF2, compris entre 0 et 100, ici normalisé entre 0 et 1) (abscisse) et son RMSD par rapport à la structure native (ordonnée). Une zone grisée dans le coininférieur droit (pLDDT >



(a) (b)

Figure 6: Corrélation entre le score de confiance (pLDDT) et la fidélité structurale (RMSD) des structures générées par RoseTTAFold2. (a) Nos calculs (b) figure S1 du preprint de RFpeptides

0.7 et RMSD < 2 Å) délimite les structures considérées à la fois comme fiables et proches de la conformation native, selon les critères utilisés dans la littérature.

On retrouve une corrélation générale entre une augmentation du pLDDT et une baisse du RMSD, ce qui est cohérent avec les observations de la figure S1. Toutefois, la répartition des points est plus étalée que dans la figure S1 du preprint de RFpeptides, et la densité dans la zone de haute qualité est un peu plus faible. Cela suggère que dans ce jeu de données, moins de structures atteignent simultanément une bonne confiance et une bonne précision.

Il est important de noter que ces résultats ont été obtenus sans recours au paramètre recycle=48, un réglage susceptible d'améliorer la qualité structurale en permettant davantage d'itérations internes durant la prédiction. Ici la valeur par défaut de recycle=3 a été utilisé, donc 16 fois moins de cycles de "recycle" (voir papier sur RF2).

4 Discussion générale

L'ensemble des résultats obtenus met en évidence le potentiel des nouvelles méthodes de design assisté par intelligence artificielle, tout en soulignant certaines limites qu'il convient d'interpréter avec nuance. Chacun des outils testés (RFdiffusion, ProteinMPNN, RoseTTAFold2) remplit un rôle bien défini dans le pipeline de design, depuis la génération d'un squelette structuré jusqu'à l'évaluation finale de sa plausibilité conformationnelle. RFdiffusion, utilisé pour construire des

échafaudages autour de motifs structuraux imposés, a montré une capacité remarquable à générer des repliements globalement cohérents. Dans le cas du motif -structuré inséré dans une architecture en hélice , l'intégration est visuellement réussie et met en valeur la force du modèle à préserver des contraintes locales. L'étape de design de séquence avec ProteinMPNN, appliquée aux structures générées, montre que les séquences obtenues ne reproduisent que partiellement celles observées dans les structures natives. Les scores calculés restent cependant cohérents avec ce que l'on attend pour un modèle génératif "de novo", et les séquences proposées s'adaptent géométriquement à leur squelette cible. Cette variabilité de séquence reflète l'existence de multiples solutions possibles à une même contrainte structurale. RosettaFold2 a été utilisé pour évaluer la faisabilité structurelle des séquences générées, en comparant les structures prédites aux modèles expérimentaux. Les RMSD obtenus varient selon les cas : faibles pour 5KWZ (0.754 Å), modérés pour 2B38 (1.553 Å) et plus élevés pour 2NB5 (1.796 Å), indiquant une précision globalement satisfaisante mais inégale. Les scores pLDDT reflètent la confiance du modèle dans ses prédictions, et une tendance se dégage : un pLDDT élevé est généralement associé à un RMSD plus faible, bien que cette corrélation ne soit pas absolue.La bonne performance observée sur certains peptides cycliques, comme 5KWZ, pourrait s'expliquer par leur présence dans le jeu d'entraînement, ce qui limiterait la portée du résultat en termes de généralisation. À l'inverse, un RMSD élevé pour 2B38 malgré un pLDDT > 0.7 suggère l'influence d'autres facteurs : taille du peptide, flexibilité locale, ou encore date de dépôt dans la PDB, indicateur possible d'inclusion dans les données d'apprentissage. Une analyse plus systématique, tenant compte de la présence dans le jeu d'entraînement et des similarités structurelles ou séquentielles, serait nécessaire pour mieux cerner les limites et biais du modèle.

5 Conclusion et perspectives

Ce stage m'a permis d'explorer le design de peptides cycliques par intelligence artificielle à travers la mise en œuvre partielle du pipeline RFpeptides. Une part importante du travail a consisté à installer, configurer et valider localement RFdiffusion, ProteinMPNN et RosettaFold2 sur le cluster du laboratoire. Ces outils, encore jamais utilisés dans l'équipe, ont nécessité une compréhension fine de leurs dépendances, de leur fonctionnement et des formats d'échange. Cette mise en place technique, préalable indispensable aux analyses, constitue une contribution significative pour l'autonomie future de l'équipe sur ce sujet. J'ai également acquis des compétences pratiques en modélisation structurale, en machine learning appliqué aux protéines, et en scripting (Python, bash, SLURM) pour l'automatisation sur cluster HPC. Plusieurs scripts sont présentés en annexe, commentés pour en faciliter la lecture. En perspective, l'intégration d'AfCycDesign permettrait de comparer ses performances à RF2 pour la validation des peptides. Une exploration plus poussée de la variabilité générée par ProteinMPNN. Enfin, ce stage m'a sensibilisé aux exigences de reproductibilité, de traçabilité et à la rigueur nécessaire pour passer d'une preuve de concept à une analyse scientifique complète.

6 Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à Dirk Stratmann, mon tuteur de stage, pour son accompagnement constant, sa bienveillance et sa disponibilité tout au long de ces mois. Il a su m'encadrer avec patience, me soutenir dans les moments plus difficiles, notamment lors de mes problèmes de santé, et m'orienter avec rigueur dans la conduite de ce projet. Son investissement humain et scientifique a été essentiel à la réussite de ce stage. Je remercie également l'ensemble de l'équipe TPM2PI (Théorie et Pratique de la Modélisation Moléculaire des Protéines et de leurs Interactions) pour leur accueil chaleureux au sein du laboratoire.

Un grand merci aussi à Zahra Fathi, camarade de stage et colocataire de bureau à RPBS, pour sa gentillesse, son écoute et son soutien émotionnel tout au long de cette expérience. Sa présence a contribué à rendre ce stage bien plus agréable au quotidien.

Enfin, je remercie tous les membres de l'UMR BFA pour l'environnement de travail stimulant et bienveillant dans lequel ce stage a pu se dérouler.

References

[1] Fanhao Wang, Tiantian Zhang, Jintao Zhu, Xiaoling Zhang, Changsheng Zhang, and Luhua Lai. Target-based *de novo* design of cyclic peptide binders, January 2025.

[2] Stephen A. Rettie, Katelyn V. Campbell, Asim K. Bera, Alex Kang, Simon Kozlov, Yensi Flores Bueso, Joshmyn De La Cruz, Maggie Ahlrichs, Suna Cheng, Stacey R. Gerben, Mila Lamb, Analisa Murray, Victor Adebomi, Guangfeng Zhou, Frank DiMaio, Sergey Ovchinnikov, and Gaurav Bhardwaj. Cyclic peptide structure prediction and design using AlphaFold2. *Nature Communications*, 16(1):4730, May 2025. Publisher: Nature Publishing Group.

[3] Xuezhi Xie, Christina Z Li, Jin Sub Lee, and Philip M Kim. CyclicBoltz1, fast and accurately predicting structures of cyclic peptides and complexes containing non-canonical amino acids using AlphaFold 3 Framework, February 2025.

[4] Stephen A. Rettie, David Juergens, Victor Adebomi, Yensi Flores Bueso, Qinqin Zhao, Alexandria N. Leveille, Andi Liu, Asim K. Bera, Joana A. Wilms, Alina Üffing, Alex Kang, Evans Brackenbrough, Mila Lamb, Stacey R. Gerben, Analisa Murray, Paul M. Levine, Maika Schneider, Vibha Vasireddy, Sergey Ovchinnikov, Oliver H. Weiergräber, Dieter Willbold, Joshua A. Kritzer, Joseph D. Mougous, David Baker, Frank DiMaio, and Gaurav Bhardwaj. Accurate *de novo* design of high-affinity protein binding macrocycles using deep learning, November 2024.

Résumé

Ce stage a porté sur l'exploration du pipeline RFpeptides, une approche récente de design de peptides cycliques basée sur l'intelligence artificielle. Trois outils principaux ont été installés, configurés et testés localement sur le cluster de calcul du laboratoire : RFdiffusion (génération de structures 3D), ProteinMPNN (design de séquences) et RosettaFold2 (validation conformationnelle). Ce travail technique, préalable indispensable à toute évaluation biologique, a permis de mettre en place une première version fonctionnelle du pipeline. Plusieurs scénarios ont été testés, incluant la génération autour de motifs structuraux, le remodelage de boucle et le design en interface. Les résultats montrent des performances variables selon les cas, soulignant l'intérêt d'une évaluation combinée (pLDDT, RMSD) pour juger de la qualité des structures générées. Ce travail constitue une première étape vers l'utilisation autonome de ces outils dans l'équipe, et ouvre la voie à des applications futures en bio-conception assistée par IA.Mots-clés : peptides cycliques, intelligence artificielle, RFpeptides, RFdiffusion, ProteinMPNN, RosettaFold2, design de novo

Abstract

This internship focused on exploring the RFpeptides pipeline, a recent AI-driven approach for designing cyclic peptides. Three core tools were installed, configured, and tested on the lab's HPC cluster: RFdiffusion (3D backbone generation), ProteinMPNN (sequence design), and RosettaFold2 (structure validation). This technical groundwork was essential for enabling further use of the pipeline. Several test scenarios were implemented, including motif scaffolding, loop remodeling, and interface-based design. The results show variable performance depending on the case, highlighting the importance of combining metrics (pLDDT, RMSD) for quality assessment. This work provides a solid basis for future applications and independent use of these tools within the team, paving the way for AI-assisted peptide engineering.Keywords: cyclic peptides, artificial intelligence, RFpeptides, RFdiffusion, ProteinMPNN, RosettaFold2, de novo design

7 Annexe

7.1 Figure S1 du preprint de RFpeptides

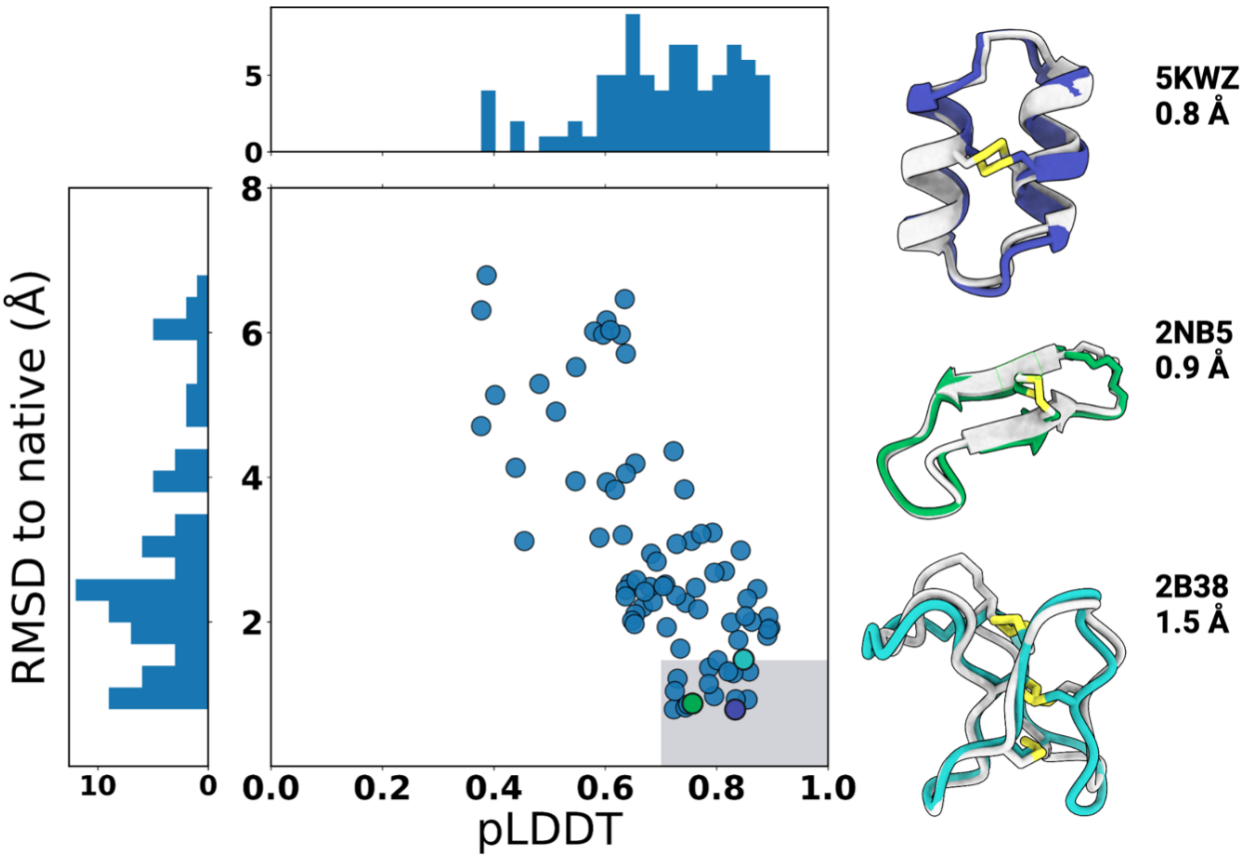


Figure 7: Figure S1 du preprint de RFpeptides[4]

7.2 Codes python/bash

Un exemple de script SLURM pour lancer un job RF2 utilisant un quart d'une carte graphique A100 (20 Go de VRAM):

```
#!/bin/bash
################################ Slurm options
   ↪ ############################

#SBATCH --job-name=2B38_48
#SBATCH --time=48:00:00
#SBATCH --partition=ipop-up
#SBATCH --gres=gpu:a100_1g.20gb:1
#SBATCH --account=cycpep_design
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --cpus-per-task=1
#SBATCH --mem=64G

#SBATCH --output=2B38_48_%j.out
#SBATCH --error=2B38_48_%j.err

#
   ↪ ##############################################################################
   ↪

module purge

apptainer exec \
  --no-mount \
  --nv \
  --bind /shared/projects/cycpep_design/software/apptainer/RoseTTAFold2/
     ↪ RoseTTAFold2:/RoseTTAFold2 \
  /shared/projects/cycpep_design/software/apptainer/RoseTTAFold2/RF2.sif
     ↪ \
  /RoseTTAFold2/run_RF2_cyclic_recy48.sh -o 2B38_RF2_recy48_out 2B38.
     ↪ fasta
```

Génération plusieurs jobs RF2 automatiquement :

#!/bin/bash
Dossiers spécifiques

BASE_DIR="/shared/projects/cycpep_design/software/apptainer/RoseTTAFold2/

FASTA_DIR="${BASE_DIR}/all_fastas" OUT_DIR="${BASE_DIR}/outputs" JOB_DIR="$
{BASE_DIR}/generated_jobs"

mkdir -p "$OUT_DIR" "$JOB_DIR"

Generation des sbatchs

for fasta in "$FASTA_DIR"/*.fasta; do base=$(basename "$fasta" .fasta) sbatch_file="$
{JOB_DIR}/try_beta_${base}.sbatch"

cat > "$sbatch_file" <<EOF
#!/bin/bash
################################# Slurm options

#SBATCH --job-name=try_beta #SBATCH --time=48:00:00
#SBATCH --partition=cmpli
#SBATCH --gres=gpu:a100_1g.20gb:1 #SBATCH --account=cycpep_design #SBATCH --nodes=1

#SBATCH --ntasks-per-node=16 #SBATCH --cpus-per-task=1 #SBATCH --mem=64G

#SBATCH --output=try_beta_${base}_%j.out #SBATCH --error=try_beta_${base}_%j.err

module purge

apptainer exec \\ --no-mount \\ --nv \\

--bind /shared/projects/cycpep_design/software/apptainer/RoseTTAFold2/ → RoseTTAFold2 :/
RoseTTAFold2 \\

/shared/projects/cycpep_design/software/apptainer/RoseTTAFold2/RF2.sif → \\

/RoseTTAFold2/run_rf2_beta.sh -o ${OUT_DIR}/${base}_RF2_out ${FASTA_DIR}/$
{base}.fasta

EOF

Soumission du job

sbatch "$sbatch_file" done

Génération du nuage de points (python) :

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import gridspec

# Charger les données
rmsd_df = pd.read_csv('/Users/rimkomputer/Desktop/m1/s2/stage/
all_rmsd.csv', sep=';')
plddt_df = pd.read_csv('/Users/rimkomputer/Desktop/m1/s2/stage/
plddt_scores.csv')  # standard , separator

# Nettoyage éventuel : retirer espaces et anomalies
rmsd_df['pdb'] = rmsd_df['pdb'].astype(str).str.strip().str.replace(',',
'').str.replace(' ', '')
plddt_df['PDB_ID'] = plddt_df['PDB_ID'].astype(str).str.strip()

# Fusionner les deux DataFrames
merged_df = pd.merge(rmsd_df, plddt_df, how='inner', left_on='pdb',
right_on='PDB_ID')

# Renommer les colonnes
merged_df = merged_df.rename(columns={'pLDDT': 'plddt'})

# Définir figure avec marges
fig = plt.figure(figsize=(8, 8))
gs = gridspec.GridSpec(2, 2, width_ratios=[4, 1], height_ratios=[1, 4],
wspace=0.05, hspace=0.05)

# Nuage de points
ax_scatter = plt.subplot(gs[1, 0])
sns.scatterplot(data=merged_df, x='plddt', y='rmsd', ax=ax_scatter)

# Délimitation d'une "zone de confiance"
ax_scatter.axvspan(0.7, 1.0, color='gray', alpha=0.2)
ax_scatter.axhspan(0.0, 2.0, color='gray', alpha=0.2)

ax_scatter.set_xlabel("pLDDT")
ax_scatter.set_ylabel("RMSD to native (Å)")

# Histogramme pLDDT (haut)
ax_histx = plt.subplot(gs[0, 0], sharex=ax_scatter)
ax_histx.hist(merged_df['plddt'], bins=20)
plt.setp(ax_histx.get_xticklabels(), visible=False)
ax_histx.set_yticks([])

# Histogramme RMSD (droite)
ax_histy = plt.subplot(gs[1, 1], sharey=ax_scatter)
ax_histy.hist(merged_df['rmsd'], bins=20, orientation='horizontal')
plt.setp(ax_histy.get_yticklabels(), visible=False)
ax_histy.set_xticks([])

plt.tight_layout()
plt.show()
```