Modélisation et score de complexes protéine-ARN

Adrien Guilhot-Gaudeffroy

UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE 427 : INFORMATIQUE PARIS-SUD

LABORATOIRE : LABORATOIRE DE RECHERCHE EN INFORMATIQUE

THÈSE

INFORMATIQUE

par

Adrien GUILHOT-GAUDEFFROY

Modélisation et score de complexes protéine-ARN

Date de soutenance : 29 / 09 / 2014

Composition du jury :

Directeurs de thèse :	M. Jérôme Azé	Professeur (LIRMM, Université de Montpellier 2)
	M^{me} Julie Bernauer	Chargée de recherche (Inria, LIX, École Polytechnique)
	M^{me} Christine Froidevaux	Professeure (LRI, Université Paris-Sud, invitée)
Rapporteurs :	M^{me} Anne Poupon	Directrice de Recherche CNRS (BIOS, INRA Tours)
	M^{me} Céline Rouveirol	Professeure (LIPN, Université Paris-Nord)
Examinateurs :	M. Philippe Dague	Professeur (LRI, Université Paris-Sud)
	M^{me} Béatrice Duval	Maître de conférences HDR (LERIA, Université d'Angers)

Sommaire

Table des figures

Liste des tableaux

Remerciements

Ce manuscrit doit son existence à l'interaction entre une multitude de personnes évoluant pour une grande part au sein de l'équipe-projet Inria AMIB (Algorithmes et Modèles pour la Biologie Intégrative), qui comprend l'équipe bioinformatique du LRI (Laboratoire de Recherche en Informatique) de l'Université Paris-Sud et l'équipe bioinformatique du LIX (Laboratoire d'Informatique de Polytechnique) de l'École Polytechnique. Je souhaite ici dresser un panorama de ces personnes, panorama qui ne se veut ni exhaustif, ni trié par score ou mérite.

J'aimerais tout d'abord remercier mes directeurs de thèse dans leur ensemble pour nos discussions scientifiques et pour leur implication au rythme de nos nombreuses réunions. Et plus spécifiquement : Julie Bernauer, pour sa minutie et ses mises en perspective ; Jérôme Azé, pour son écoute et ses propos rassurants ; Christine Froidevaux, pour sa rigueur et sa supervision tout au long des différentes étapes de la thèse. Ce sont mes encadrants qui ont contribué à me former sur ce qu'aucun manuel n'a su me transmettre.

Je tiens aussi à remercier l'équipe bioinfo AMIB au complet pour cette ambiance de travail à la fois studieuse et agréable. Je remercie aussi plus particulièrement les doctorants de l'équipe bioinfo AMIB pour leur soutien en cas de besoin et leurs sujets de discussion passionnants sans cesse renouvelés.

Merci aussi à Sid Chaudhury et Jeff Grey pour leurs conseils avisés en matière de prédiction structurale.

Je remercie Philippe Dague et Béatrice Duval pour avoir accepté de faire partie de mon jury de thèse, ainsi qu'Anne Poupon et Céline Rouveirol pour avoir aussi accepté d'être les rapporteurs de ce manuscrit.

Un grand merci aux équipes techniques du LRI et du LIX pour leur efficacité et leur aide au jour le jour sur le support informatique, ainsi que les équipes administratives du LRI comme du LIX et le secrétariat de l'École doctorale pour leur patience.

Je remercie l'équipe de communication d'Inria pour leur sympathie et les journées de vulgarisation très instructives. Merci aussi au SIP (Service d'Insertion Professionnelle) pour ses formations sur l'écriture de textes scientifiques et la prise de parole. Et merci à Michèle Sebag pour ses cours d'apprentissage automatisé.

Je souhaiterais également remercier Anne Fourier, pour sa compréhension, ainsi que son soutien inconditionnel durant les nuits blanches – ou presque – improvisées et les réveils difficiles associés.

Je tiens aussi à signaler qu'une grande partie des calculs nécessaires aux résultats évalués au cours de cette thèse n'auraient pas vu le jour en moins de trois ans sans la participation de certaines ressources de calcul. Parmi ces ressources se trouvent le cluster de l'équipe bioinfo AMIB, celui du LRI, ainsi que les ressources de CHP (Calcul Haute Performance) du TGCC (Très Grand Centre de calcul du CEA), l'allocation t2013077065 attribuée par Genci (Grand Équipement National de Calcul Intensif).

Enfin, je remercie le Ministère de l'Enseignement Supérieur et de la Recherche ainsi que toutes les personnes qui ont contribué à me fournir la bourse doctorale, sans laquelle cette thèse n'aurait pas débuté.

Introduction

1 Contexte

Les complexes protéine-ARN jouent un rôle majeur dans la cellule. Ils sont impliqués dans de nombreux processus cellulaires tels que la réplication et la transcription des ARN messagers. Leur régulation est souvent clef dans les mécanismes qui mettent en jeu de larges machineries moléculaires (comme le RISC, *RNA-Induced Silencing Complex*) impliquées dans les cancers. Les interactions protéine-ARN présentent donc un grand intérêt pour les études à visée thérapeutique [60]. Les protéines étant capables d'interagir avec l'ARN sont nombreuses et variées : leur structure met en œuvre de nombreux domaines structuraux. Entre autres, les domaines RRM et dsRDB montrent tous une activité de liaison à l'ARN et sont très étudiés [49]. Ces dernières années, les techniques expérimentales de résolution ont mis en évidence de nouvelles structures d'ARN et de complexes protéines-ARN. Grâce à la cristallographie [133] et à la résonance magnétique nucléaire (RMN) [227, 239], nous disposons aujourd'hui de structures à haute résolution qui permettent de mieux comprendre les fonctions des ARN et leurs modes d'association [44, 78]. D'autres méthodes expérimentales permettent quant à elles d'étudier à basse résolution des structures beaucoup plus grandes [138, 166, 179]. Les expériences sur les molécules uniques fournissent même des données à haute résolution [270] et la conception de molécules d'ARN est désormais accessible [45]. Malgré ces avancées majeures en biologie structurale pour les ARN et les complexes protéine-ARN, le nombre de structures disponibles dans la *Protein Data Bank*, base de données expérimentale de référence, reste faible : de l'ordre d'un ou deux milliers. La modélisation et la prédiction de ces complexes sont donc nécessaires, bien que difficiles [203].

La modélisation des assemblages moléculaires est un problème complexe et pour lequel de nombreuses méthodes de prédiction et d'évaluation des résultats ont été développées [173, 181, 249]. Le challenge international CAPRI *(Critical Assessment of PRediction of Interactions)*[1] [121], qui évalue les prédictions faites à l'aveugle, a montré que, malgré de grands progrès, les méthodes actuelles d'amarrage ont besoin d'une grande quantité et d'une grande variété de données expérimentales [67]. Bien que les techniques récentes soient capables de mieux prédire et prendre en compte les ions et molécules d'eau qui interviennent dans les interactions [146], la flexibilité des molécules reste un frein majeur. Même s'il n'est pas aujourd'hui possible de prédire

1. `http://capri.ebi.ac.uk`

les affinités de liaison entre molécules, l'originalité et la qualité des premiers résultats obtenus sont encourageantes [146].

Les interactions entre protéines et ARN sont difficiles à prédire, principalement pour deux raisons : la flexibilité des molécules d'ARN d'une part et les forces électrostatiques qui guident l'interaction des molécules d'ARN chargées négativement d'autre part. Les progrès récents des techniques de prédiction de structure d'ARN et de repliement [65, 66, 142, 195, 217] permettent de prendre en compte la flexibilité de l'ARN, mais le plus souvent à l'échelle atomique uniquement [85] et ne permettent pas aisément l'intégration dans une procédure d'amarrage. Cela ne pourra être fait qu'en disposant de fonctions de score assez performantes pour sélectionner les bonnes conformations. Les adaptations gros-grain qui permettent de réduire notoirement les phases initiales de recherche sont intéressantes [153, 229] et les champs de forces statistiques dédiés [43, 113, 205, 247, 269] sont prometteurs. Toutefois, ces optimisations sont souvent basées sur de simples mesures et font peu usage des jeux de données à valeur ajoutée disponibles dans la communauté. La *Protein-RNA Interface DataBase* (PRIDB) [152] fournit des jeux de données nettoyés de qualité. Disponibles à différents niveaux de redondance, ils permettent des mesures atomiques précises. Les trois jeux d'essais conçus pour l'amarrage protein-ARN et disponibles dans la littérature [9, 112, 204] permettent d'évaluer les prédictions.

Pour l'amarrage, la disponibilité de données structurales est fondamentale pour les méthodes d'apprentissage. Plusieurs méthodes issues de l'apprentissage ont été développées pour les complexes protéine-protéine [6, 18, 15, 22, 27]. Ces méthodes se sont avérées efficaces pour la réévaluation *a posteriori (rescoring)* et l'optimisation d'expériences d'amarrage *in silico*, comme l'ont montré les dernières éditions de CAPRI [251, 272]. Les techniques d'apprentissage sont donc de plus en plus étudiées et se sont développées au sein de la communauté CAPRI [148].

Dans ce travail, je me suis intéressé au développement de nouvelles méthodes d'apprentissage pour les complexes protéine-ARN. Dans cette introduction, après avoir présenté la structure des différentes molécules mises en jeu ainsi que leurs spécificités structurales, je détaillerai les différents modèles de représentation disponibles. Ensuite, je décrirai le principe des méthodes d'amarrage et de score. Et j'aborderai enfin les principes et méthodes d'apprentissage en lien avec le problème de l'amarrage.

2 Structure des protéines et des ARN

Une protéine est une macromolécule biologique constituée d'un enchaînement linéaire d'acides aminés reliés par une liaison peptidique. La protéine est le résultat de l'expression d'un gène, porté par l'acide désoxyribonucléique (ADN), qui est d'abord transcrit en acide ribonucléique (ARN) messager, lui-même traduit en protéine par le ribosome (voir fig. 1). Les ARN sont formés d'un enchaînement linéaire d'acides nucléiques. De nombreux ARN ne codent pas pour des protéines et ont une structure et une fonction propre. On distingue par exemple les ARN de tranfert (ARNt). Leur fonction est de permettre l'ajout d'un acide aminé à la chaîne d'acides aminés d'une

protéine en cours de synthèse. Les ARNt ont une structure très particulière, dite en trèfle du fait de leur forme en représentation 2D.

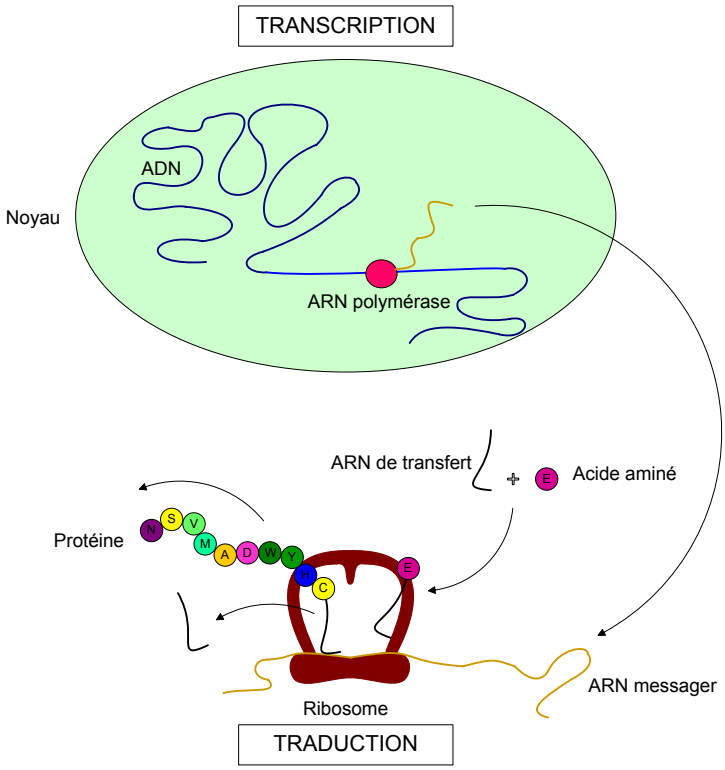


FIGURE 1 – Du gène à la protéine. *Le gène (en bleu clair), appartenant à une molécule d'ADN, est transcrit en ARN messager (en beige) par un enzyme : l'ARN polymérase. Puis l'ARN messager est traduit en protéine dans le ribosome. À cette traduction participent notamment les ARN de transfert. Image de J. Bernauer.*

Dans les conditions physiologiques, la protéine se replie, adoptant par ce biais une conformation compacte spécifique. Ce repliement peut être spontané – par l'interaction de la protéine avec le solvant – ou bien se faire grâce à des protéines spécialisées, appelées les chaperonnes. Dans certains cas, une maturation peut se produire par l'ajout de glucides ou bien par le clivage de certaines parties de la protéine. Ensuite, la protéine, mature et repliée, est soit libérée dans le cytoplasme, soit dirigée vers une membrane, soit encore excrétée dans le milieu extérieur.

Tout changement, qu'il soit dans les conditions du milieu (température, pH, force ionique, présence d'agents chimiques) ou dans l'enchaînement des acides aminés (par mutagenèse ou par modification chimique), peut modifier le repliement de la protéine ou ses interactions et moduler son activité.

Le repliement des protéines est imposé d'une part par les interactions entre les différents acides aminés qui les composent et, d'autre part, entre ces acides aminés et le solvant. Cependant, toutes les interactions ne sont pas parfaitement connues

et la complexité de ce phénomène est telle que, à l'heure actuelle, son déroulement est impossible à décrire. La structure adoptée par une protéine de même que ses interactions avec différents partenaires peuvent être déterminées expérimentalement mais cette détermination peut être longue et difficile, même avec l'essor des projets de génomique structurale qui ont mis en place des stratégies de résolution de structure à haut débit. Étant donné le nombre actuellement connu de séquences protéiques et d'interactions potentielles, la détermination expérimentale de toutes les structures correspondantes n'est pas envisageable. Il faut donc se tourner vers la modélisation, qui tente de prévoir non seulement le repliement des protéines en partant de leurs séquences, mais aussi la conformation du complexe à partir de la structure de chacun de ses partenaires déterminée séparément.

De la même façon, le repliement des ARN est dû à l'interaction entre les nucléotides qui le composent, le solvant et – les molécules d'ARN étant chargées – les ions. Comme pour les protéines, les interactions mises en jeu lors du repliement sont mal connues. La connaissance de ces interactions est rendue encore plus difficile par les aspects électrostatiques et les liaisons hydrogène très nombreuses. Les molécules d'ARN sont à la fois beaucoup plus flexibles et moins stables que les protéines, ce qui rend à la fois leur résolution expérimentale et leur modélisation plus difficiles.

On définit plusieurs niveaux d'organisation de la structure des protéines et des acides nucléiques (voir fig. 2) :

- la structure primaire, ou séquence, est l'enchaînement des acides aminés ou des nucléotides ;
- la structure secondaire résulte d'interactions à courte distance (essentiellement des liaisons hydrogène entre atomes). Pour les protéines, elles ne dépendent qu'en partie de la nature des chaînes latérales des acides aminés impliqués : certains segments de la protéine adoptent ainsi une conformation périodique d'angles dièdres successifs. Pour les ARN, ces interactions sont liées à la nature des nucléotides et au type d'appariement que ceux-ci peuvent former ;
- la structure tertiaire est la forme fonctionnelle repliée d'une chaîne protéique ou nucléique. Elle résulte de l'assemblage selon une topologie déterminée des structures secondaires ;
- la structure quaternaire, qui comprend les complexes, est l'association de plusieurs chaînes d'acides aminés ou d'acides nucléiques (de chaînes identiques ou non).

2.1 Les acides aminés

Un acide aminé est une molécule constituée d'un carbone asymétrique (appelé carbone α ou C_α) lié à un groupement carboxyle COOH, un groupement aminé NH_2, un hydrogène H et un radical R, aussi appelé la chaîne latérale.

Selon la conformation du carbone α, on parle d'acide aminé D ou L (voir fig. 3).

Les appellations D et L ont été données à l'origine pour désigner les composés dextrogyres et lévogyres. Cependant, la conformation D ou L ne permet pas de prévoir les propriétés optiques d'un acide aminé. Chaque acide aminé doit son nom à la nature de son radical. Les acides aminés naturels les plus courants sont au nombre de 20,

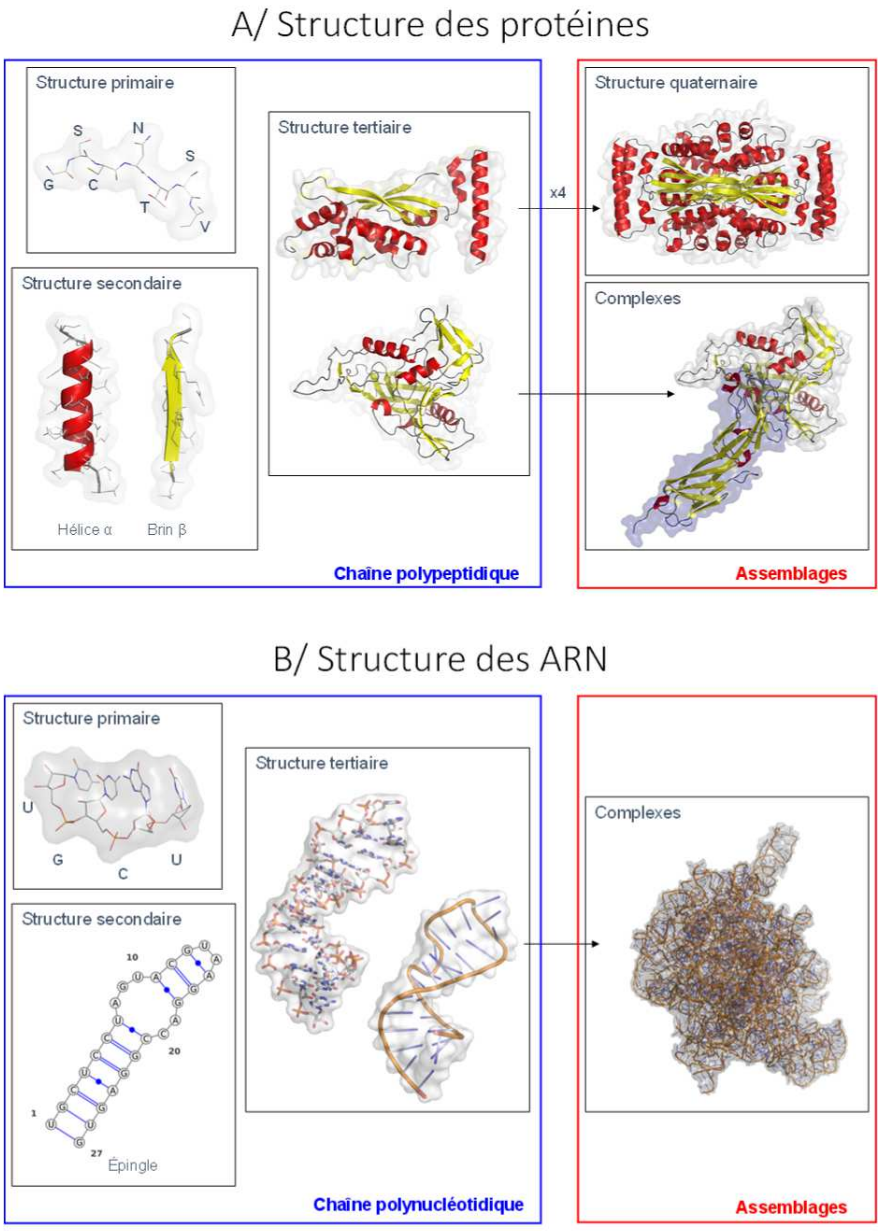
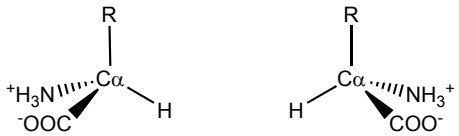


FIGURE 2 – Les différents niveaux de structure : (A) protéines, (B) ARN. La fonction des assemblages formés est dépendante des différents niveaux. Image de J. Bernauer.



L-acide aminé D-acide aminé

FIGURE 3 – Formes D et L d'un acide aminé. Ces formes, non superposables, sont l'image l'une de l'autre dans un miroir. Image de J. Bernauer.

tous de configuration L (voir fig. 4). Certaines protéines contiennent un petit nombre d'acides aminés modifiés tels que l'hydroxyproline ou la sélénocystéine.

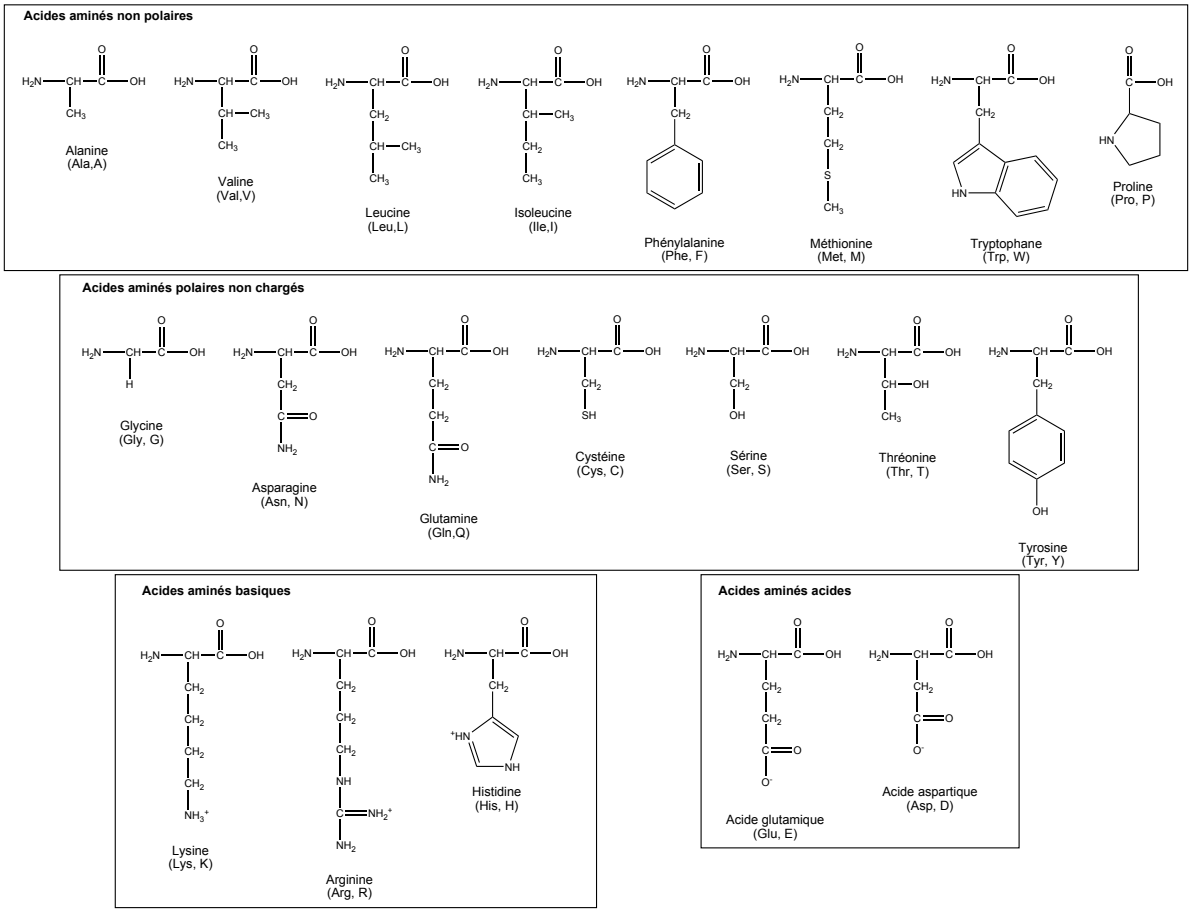


FIGURE 4 – Les 20 acides aminés usuels. Image de J. Bernauer.

La nature du radical R (aussi appelé chaîne latérale) confère à chaque acide aminé des propriétés physico-chimiques particulières (encombrement stérique, hydrophobie, polarité, acidité, flexibilité, *etc.*). Ces propriétés permettent le repliement de la protéine, garantissent ainsi sa stabilité, et permettent son activité biochimique.

2.2 Les acides nucléiques

Un acide nucléique, comme l'ARN ou l'ADN, est formé de sous-unités appelées des nucléotides. Les nucléotides sont formés d'une base azotée appelée une base, d'un cycle à cinq atomes de carbone appelé un sucre (ribose pour l'ARN et désoxyribose pour l'ADN) et d'un groupement phosphate (voir fig. 5). Comme pour les acides aminés, pour les nucléotides, les parties correspondant au squelette – à savoir le sucre et le groupement phosphate – sont identiques pour tous les nucléotides. Ceux-ci ne diffèrent donc que par leurs bases.

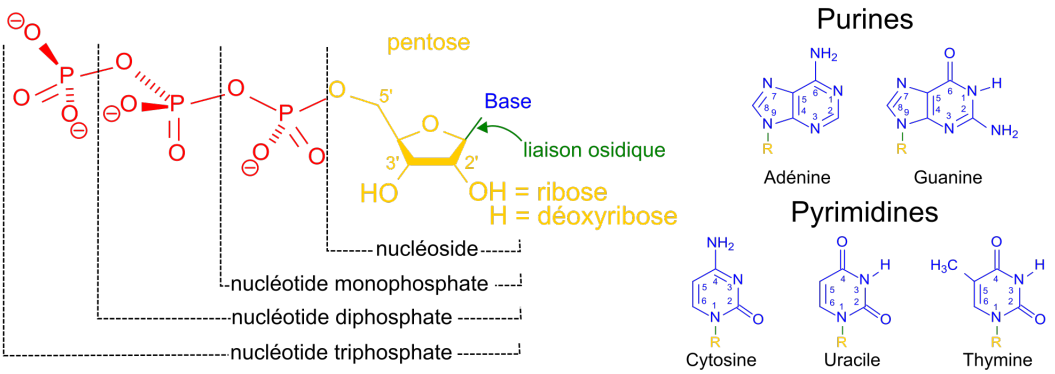


FIGURE 5 – Les nucléotides. Image de Wikipedia.

On distingue principalement cinq types de bases : l'adénine (A), la guanine (G), la cytosine (C), l'uracile (U) et la thymine (T). Elles appartiennent à deux familles différentes : les purines, qui sont formées de deux cycles aromatiques (l'un à cinq et l'autre à six atomes), et les pyrimidines qui sont formées d'un cycle aromatique à six atomes. L'ARN contient principalement les quatre bases A,U,G et C. Il existe aussi d'autres nucléotides parfois en interaction dans un complexe protéine-ARN : ce sont les nucléotides non standards. Ces nucléotides non standards sont généralement obtenus par l'ajout ou le retrait d'un groupement chimique à la base azotée.

2.3 La structure primaire ou séquence

2.3.1 Définition

La structure primaire d'une protéine, également appelée chaîne polypeptidique, est l'enchaînement de ses acides aminés. Lors de la traduction, le groupement acide d'un acide aminé est lié au groupement aminé de l'acide aminé suivant ; cette liaison est appelée liaison peptidique. La nature de cette liaison impose certaines contraintes spatiales : en particulier, le C et le O du groupement carboxyle du premier acide aminé, ainsi que le N et le C_α de l'acide aminé suivant sont coplanaires (voir fig. 6).

Les liaisons covalentes établies lors de la traduction ne sont généralement pas modifiées. Les exceptions peuvent être la coupure de certaines parties de la chaîne protéique, l'établissement de ponts disulfure entre deux cystéines ou encore la liaison avec des glucides lors de la maturation. Une protéine comprend entre 30 et 30 000 acides aminés, la moyenne se situant autour de 330 [268]. On appelle squelette de la protéine, la chaîne des N, C_α, C, et O de tous les acides aminés qui la constituent.

La structure primaire d'un ARN est la succession de ses nucléotides. Les bases étant formées de cycles aromatiques, elles sont planes. Les bases s'apparient de la façon suivante : l'adénine forme deux liaisons hydrogène avec l'uracile et la guanine forme trois liaisons hydrogène avec la cytosine (voir fig. 7). Cette dernière paire est donc plus stable.

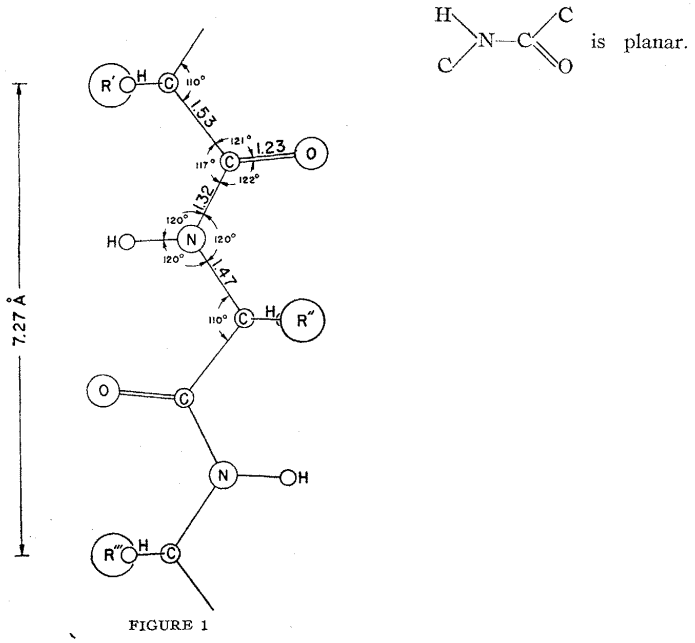


FIGURE 6 – Géométrie de la liaison peptidique telle que décrite par Linus Pauling [197], figure 1.

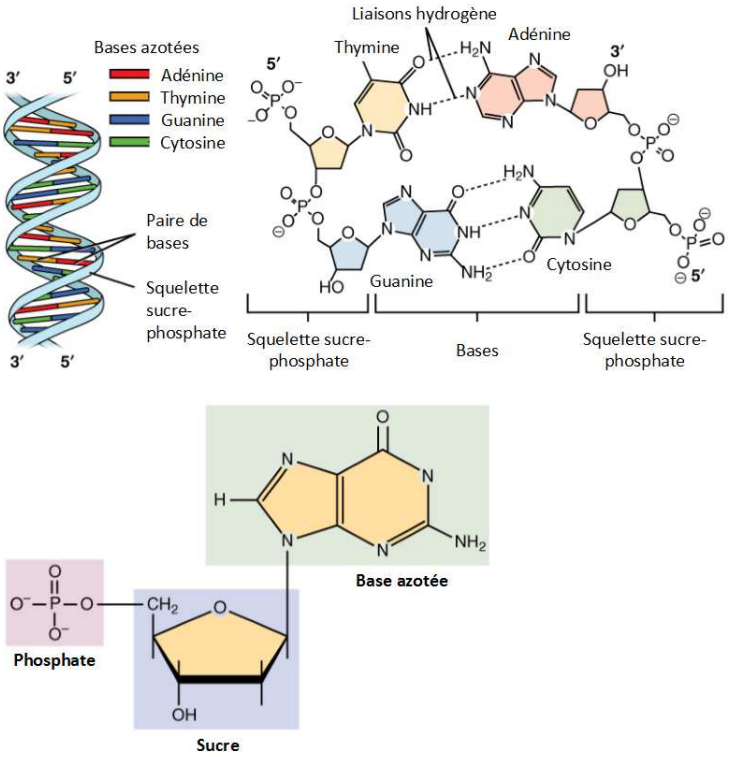


FIGURE 7 – Appariement des nucléotides et hélice d'ADN. Image adaptée de Wikipedia.

2.3.2 Détermination

La détermination de la séquence d'une protéine est une étape indispensable de son étude. En effet, la séquence donne non seulement des informations sur le repliement de la protéine (surtout s'il est possible de détecter des similitudes avec des protéines dont la structure est connue), mais également sur sa fonction et sur sa localisation cellulaire.

Cette détermination se fait généralement de manière indirecte, par traduction de la séquence du gène. De nombreuses techniques peuvent être utilisées à des fins de séquençage à grande échelle [111].

Cette méthode ne permet cependant pas de connaître d'élements tels que les événements post-traductionnels, en particulier les substitutions, les glycosylations ou les délétions d'une partie de la chaîne. Pour connaître directement la structure primaire d'une protéine de petite taille (moins de 150 acides aminés), il est possible d'utiliser la spectroscopie de masse [237]. Mais la séquence peut également être découpée – par digestion enzymatique ou chimique – en tronçons d'une longueur inférieure à 30 acides aminés. La séquence de l'ensemble de ces tronçons peut ensuite être déterminée par micro-séquençage.

La détermination d'une séquence d'ARN se fait par séquençage, soit directement, soit par conversion en ADN complémentaire (ADNc) à l'aide d'une transcriptase inverse.

2.4 La structure secondaire

2.4.1 Définition

On appelle structure secondaire régulière une partie de la chaîne adoptant une conformation périodique. Ces structures sont stabilisées par des réseaux de liaisons hydrogène entre les acides aminés non voisins dans la chaîne polypeptidique.

Les premières structures secondaires ont été définies par Linus Pauling en 1951 [196, 197] : l'hélice α et le brin β. Ce sont deux structures secondaires très largement majoritaires dans les protéines. Ces organisations moléculaires d'une part minimisent les gênes stériques et les répulsions électrostatiques entre les chaînes latérales et maximisent le nombre de liaisons hydrogène : elles sont donc très largement favorisées [128].

Dans une protéine, en moyenne, la moitié des acides aminés est impliquée dans des structures secondaires régulières. L'autre moitié des résidus se trouve impliquée dans des boucles reliant entre elles les structures secondaires régulières. En moyenne, la longueur d'un brin β est de cinq acides aminés, celle d'une hélice α est de six acides aminés, tandis que celle d'une boucle est de douze acides aminés [51].

L'hélice α est stabilisée par des liaisons hydrogène entre des acides aminés de la même hélice – des acides aminés distants de seulement $3,5$ résidus en moyenne dans la chaîne polypeptidique (voir fig. 8). Même isolée, l'hélice α est stable.

Au contraire, le brin β n'est stable qu'associé à au moins un autre brin β, formant ainsi un feuillet (voir fig. 9). Les brins d'un feuillet peuvent être tous dans le même sens

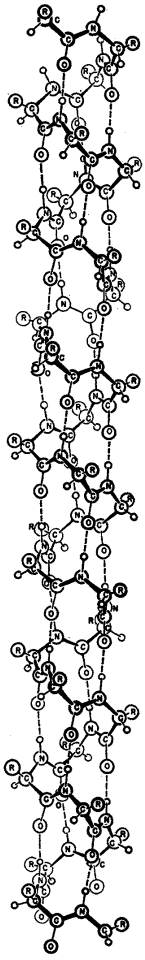


FIGURE 2
The helix with 3.7 residues per turn.

FIGURE 8 – Hélice α telle que représentée dans l'article original de Linus Pauling [197], figure 2.

– on parle alors de brins parallèles – ou bien être positionnés alternativement dans un sens et dans l'autre – on parle alors de brins antiparallèles. Les liaisons hydrogène

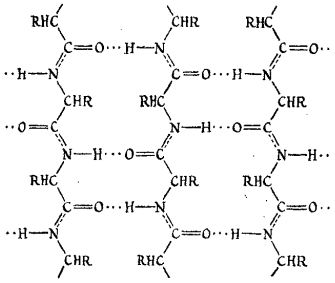


FIGURE 1

Diagrammatic representation of a hydrogen-bonded layer structure of polypeptide chains with alternate chains oppositely oriented.

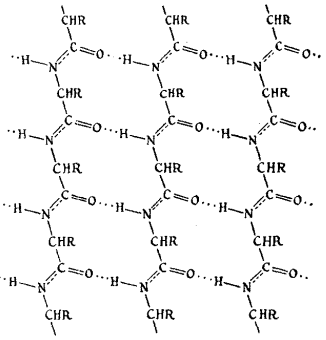


FIGURE 2

Diagrammatic representation of a hydrogen-bonded layer structure of polypeptide chains with all chains similarly oriented (the pleated sheet).

FIGURE 9 – Brins β tels que représentés dans l'article original de Linus Pauling [196].

assurant la cohésion des acides aminés entre eux s'établissent entre acides aminés distants dans la séquence.

La quantité et l'agencement des structures secondaires régulières conduisent à classer les protéines en cinq catégories : tout-α, tout-β, α/β (alternance d'hélices α et de brins β), $\alpha+\beta$ (structures contenant des hélices α et des brins β sans alternance), et "autres" [185, 191].

La structure secondaire des ARN correspond aux motifs créés par les différents appariements des nucléotides. Alors que l'ADN existe principalement sous forme de doubles hélices complètement appariées, l'ARN est souvent simple brin et forme de nombreux motifs variés. L'ARN est en effet plus flexible et peut former des structures plus complexes, du fait des liaisons hydrogène possibles avec le groupement hydroxyle du sucre.

On distingue plusieurs motifs de structures secondaires pour l'ARN : les hélices et différents types de boucles. Les enchaînements possibles de ces éléments sont parfois classés en familles de structures secondaires : les tetraloops, les pseudonœuds, les tiges-boucles, *etc*.

Bien qu'étant un motif de structure tertiaire, pour les acides nucléiques, l'hélice dépend de la structure secondaire : en effet, elle correspond à une région de structure secondaire formée de paires de bases consécutives.

La tige-boucle est un motif qui correspond à une hélice terminée par une courte boucle de nucléotides non appariés (voir fig. 10). C'est un motif extrêmement courant et qu'on retrouve dans des motifs plus grands tels que le trèfle à quatre feuilles qui caractérise les ARN de transfert. Les boucles internes, série de bases non appariées au sein d'une hélice, et les renflements, régions dans lesquelles un brin est composé de bases insérées "en plus" non appariées sont aussi fréquentes. Ces régions sont aussi parfois appelées jonctions.

Les pseudonœuds correspondent à une structure secondaire disposant de deux

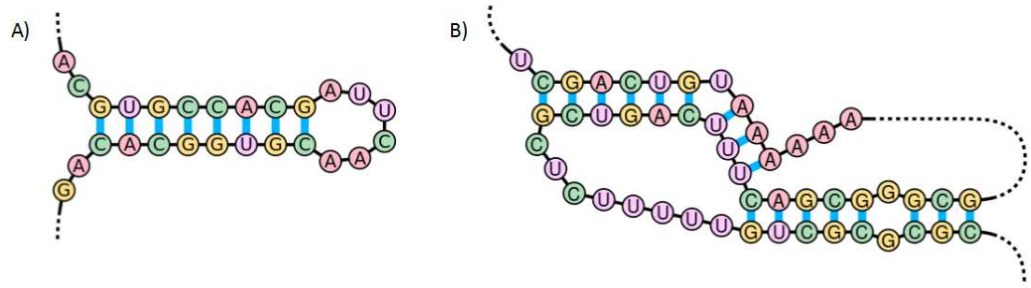


FIGURE 10 – Exemple de motifs de structure secondaire d'ARN : A) la tige boucle B) le pseudonœud. Image adaptée de Wikipedia.

tiges-boucles et dans laquelle la moitié d'une tige est intercalée entre les deux moitiés de l'autre tige (voir fig. 10). Les pseudonœuds se replient en 3D en forme de nœuds mais ne sont pas des nœuds au sens topologique. De multiples processus biologiques reposent sur la formation de pseudonœuds (l'ARN de la télomérase humaine par exemple). Bien que l'ADN puisse former des pseudonœuds, ceux-ci ne sont trouvés naturellement que chez les ARN.

2.4.2 Détermination à partir d'une solution

À partir d'une solution de protéine purifiée, il est possible d'estimer la composition globale en structures secondaires régulières (nombre d'acides aminés participant à des brins β ou à des hélices α) par des méthodes spectroscopiques :

- dichroïsme circulaire vibrationnel ou ultra-violet [194] ;
- spectroscopie infrarouge [202] ;
- spectroscopie Raman [4] ;
- analyse des déplacements chimiques en RMN (résonance magnétique nucléaire) [258].

Cependant, pour les protéines, l'unique moyen de déterminer avec précision la position dans la structure tertiaire de ces structures secondaires régulières reste de déterminer complètement la structure tertiaire. Pour l'ARN, la structure secondaire est en plus accessible grâce à la structure tertiaire, mais celle-ci est bien plus difficile à résoudre expérimentalement. Il est toutefois possible d'obtenir la structure secondaire de façon expérimentale : soit par séquençage, soit à l'aide de méthodes de sondage. On peut citer les sondages par modification chimique utilisant :

- les radicaux hydroxyles, qui attaquent les cycles des sucres exposés [129, 246] ;
- le DMS (dimethyl sulfate), qui modifie certaines bases en les méthylant, et les sites qui ne peuvent plus ensuite s'apparier sont détectés par RT-PCR [241] ;
- le CMCT (1-Cyclohexyl-3-(2-Morpholinoethyl)Carbodiimide metho-p-Toluene sulfonate), qui modifie les uridines et les guanines exposées (suivi aussi d'une détection par RT-PCR) [88] ;
- le kethoxal (1,1-Dihydroxy-3-ethoxy-2-butanone), qui modifie aussi les guanines exposées [97] ;
- et la méthode de sondage SHAPE *(Selective 2'-Hydroxyl Acylation analyzed by*

Primer Extension), qui comprend des réactifs ayant une préférence pour les zones flexibles du squelette de l'ARN [176].

Il existe aussi des méthodes de sondage sans traitement chimique *(In-line probing)* qui permettent de voir les changements structuraux dus aux interactions [97] ou de cartographie par interférence utilisant des analogues de nucléotides *(NAIM)* [219].

2.4.3 Détermination à partir des coordonnées atomiques

Même lorsqu'on dispose des coordonnées atomiques d'une protéine, il n'est pas évident d'identifier les structures secondaires régulières. Bien évidemment, il ne s'agit plus ici de déterminer le nombre et la nature des structures secondaires régulières, mais plutôt de déterminer la position exacte de leurs extrémités dans la séquence. Il existe de nombreux programmes permettant de réaliser l'attribution des structures secondaires des protéines, à savoir : dire à quel type de structure secondaire participe chaque acide aminé. La comparaison de ces programmes montre que les résultats obtenus par les différentes méthodes peuvent être assez différents au niveau des limites de chaque structure secondaire.

Pour les ARN, la détermination des structures secondaires est bien plus simple. L'attribution d'une structure secondaire aux acides nucléiques des extrémités peut être délicate, mais de nombreux programmes proposent cette détermination.

2.4.4 Prédiction

La prédiction des structures secondaires est une étape intéressante de l'étude d'une protéine. En effet, elle permet d'émettre des hypothèses sur la nature du repliement, aide à localiser des résidus du site actif, ou encore à donner une hypothèse quant à la localisation de la protéine dans la cellule (en particulier pour les protéines membranaires).

Il existe un certain nombre de logiciels de prédiction de structure secondaire, fondés sur des méthodes différentes [87, 216]. Désormais, les prédictions obtenues sont exactes à plus de 75 %, comme le montrent les résultats de l'expérience CASP.

De la même façon, la prédiction des structures secondaires est un champ de recherche très actif. De nombreuses techniques ont été développées [110, 171, 210, 230, 274]. Une des difficultés majeures est ensuite de savoir dans quelle mesure les structures locales des bases affectent la structure tertiaire des ARN.

2.5 La structure tertiaire

2.5.1 Définition

La structure tertiaire d'une protéine est la description du repliement d'une chaîne polypeptidique en sa forme fonctionnelle, ainsi que des liaisons covalentes apparues après la traduction (essentiellement les ponts disulfure), la présence éventuelle d'ions ou de cofacteurs plus complexes (hème, flavine adénine dinucléotide ou FAD...). Les structures tertiaires sont très variées et très complexes.

Les chaînes polypeptidiques de grande taille (plus de 200 acides aminés) se replient souvent en plusieurs régions fonctionnelles. On parle de domaine si ces unités fonctionnelles adoptent un repliement stable lorsqu'elles sont isolées.

Quand deux séquences protéiques présentent plus de 30 % d'identité de séquence, elles adoptent le même repliement [47, 221]. En dessous de ce seuil, il est difficile de prévoir, par les méthodes classiques d'alignement de séquence, si deux protéines vont adopter la même structure tertiaire. De plus, certaines protéines adoptent des repliements similaires sans présenter d'identité de séquence détectable ; c'est le cas notamment de la superfamille des immunoglobulines [102].

Le repliement repose principalement sur des interactions à courte distance. Ces interactions ont lieu, d'une part, entre les acides aminés enfouis dans la protéine, et, d'autre part, entre les acides aminés de la surface et les molécules du solvant [215]. Ces interactions sont des liaisons hydrogène, des ponts salins ou des liaisons de type Van der Waals.

La structure tertiaire des ARN est, de la même façon, la description du repliement de la chaîne polynucléotidique en sa forme 3D fonctionnelle. Celle-ci repose principalement sur les appariements Watson-Crick (GC et AU) qui forment les hélices. Dans les acides nucléiques, les hélices sont des polymères en forme de spirale, en général droite, contenant deux brins de nucléotides appariés. Un tour d'hélice est constitué d'environ 10 nucléotides et contient un grand et un petit sillon. Étant donné la différence de largeur entre le petit et le grand sillon, de nombreuses protéines se lient par le grand sillon. De nombreux types d'hélices sont possibles : pour l'ARN, on rencontre principalement des hélices A.

2.5.2 Détermination

La première structure de protéine résolue a été celle de la myoglobine [134] par cristallographie aux rayons X. À l'heure actuelle, la *Protein Data Bank* (PDB) [13, 14], banque de données des structures tridimensionnelles des protéines, contient plus de 33 000 fichiers, dont environ 28 000 correspondent à des structures résolues par cristallographie et 5 000 à des structures résolues par RMN (dans sa version *PDB 2004 archives release #1*). D'autres méthodes de résolution de structure peuvent aussi être utilisées, mais elles restent pour l'instant moins efficaces.

Ces méthodes, même si leurs performances se sont beaucoup améliorées, en particulier avec l'apparition des projets de génomique structurale, restent tributaires de conditions expérimentales restrictives. La cristallographie nécessite l'obtention de cristaux diffractants, ce qui demande beaucoup de matériel et de travail. Quant à la RMN, même si la contrainte du cristal est supprimée, elle ne peut s'appliquer que sur des protéines relativement petites (moins de 300 résidus) et il faut obtenir une quantité importante de solution de protéine pure à plus de 95 %. Étant donné le nombre de séquences connues à l'heure actuelle, il n'est donc pas envisageable de résoudre toutes les structures correspondantes.

Par exemple pour la cristallographie X, selon la protéine et la qualité du cristal, on connaît la structure avec une résolution plus ou moins bonne. À basse résolution (supérieure à 3 Å), on connaît le squelette de la protéine et les structures secondaires.

À moyenne résolution, on peut observer les interactions entre acides aminés, en particulier les liaisons hydrogène et les interactions de type Van der Waals. À haute résolution (moins de 1.5 Å), on peut déterminer avec précision longueurs et angles des liaisons, l'hydratation, et les mouvements atomiques autour des positions d'équilibre.

Ces méthodes sont aussi utilisées pour les ARN, mais ceux-ci sont plus flexibles et beaucoup moins stables, ce qui rend leur résolution beaucoup plus complexe. Alors que la *Protein Data Bank* contient aujourd'hui plus de 100 000 structures, quelques milliers d'entre elles seulement correspondent à des structures d'ARN.

2.5.3 Prédiction

Modélisation par homologie Lorsqu'on peut établir une similitude entre la séquence dont on cherche la structure et une séquence dont la structure tridimensionnelle est connue *(support)*, il est possible de construire un modèle de la structure recherchée.

Un modèle obtenu de la sorte est d'autant plus précis que l'identité de séquence entre le support et la séquence à modéliser est forte. Pour de faibles taux d'identité, on ne connaît avec précision, dans le modèle, que les acides aminés strictement conservés, et les parties ne comportant pas de longues insertions/délétions. Le modèle obtenu n'est donc pas l'équivalent d'une structure déterminée par des méthodes physiques. Cependant, il rend souvent compte du comportement du site actif ou encore des parties de la protéine nécessaires à son repliement ou à son interaction avec des partenaires.

Les méthodes d'enfilage Les méthodes d'enfilage, ou *threading*, permettent de tester la compatibilité d'une séquence avec un repliement [240]. Dans ce cas, pour une séquence donnée, on cherche parmi les structures connues, celle qui est la plus compatible avec la séquence dont on dispose.

La modélisation *ab initio* La finalité des techniques de modélisation *ab initio* est de prédire la structure d'une protéine à partir de sa seule séquence. De nombreux modèles de calculs sont utilisés, faisant appel par exemple à la dynamique moléculaire. Mais, même si les progrès sont conséquents, les résultats sont très variables, comme l'atteste l'expérience *CASP* [25] ou l'état du projet *folding@home*[2].

Évaluation des prédictions : l'expérience CASP[3] L'expérience CASP *(Critical Assessment of Methods of Protein Structure Prediction)* est une compétition qui a lieu tous les deux ans depuis 1994 et a pour objectif de tester les méthodes de prédiction de structure. Des protéines, dont la structure vient d'être résolue mais pas encore publiée, sont proposées aux prédicteurs. Ceux-ci doivent tenter de prédire, selon la catégorie, la structure *ab initio*, la structure par homologie ou la structure secondaire.

2. http://folding.stanford.edu/
3. http://predictioncenter.org/

Les évaluations des prédictions [62, 141, 182] montrent d'importants progrès dans la prédiction de structure *ab initio* (voir fig. 11).

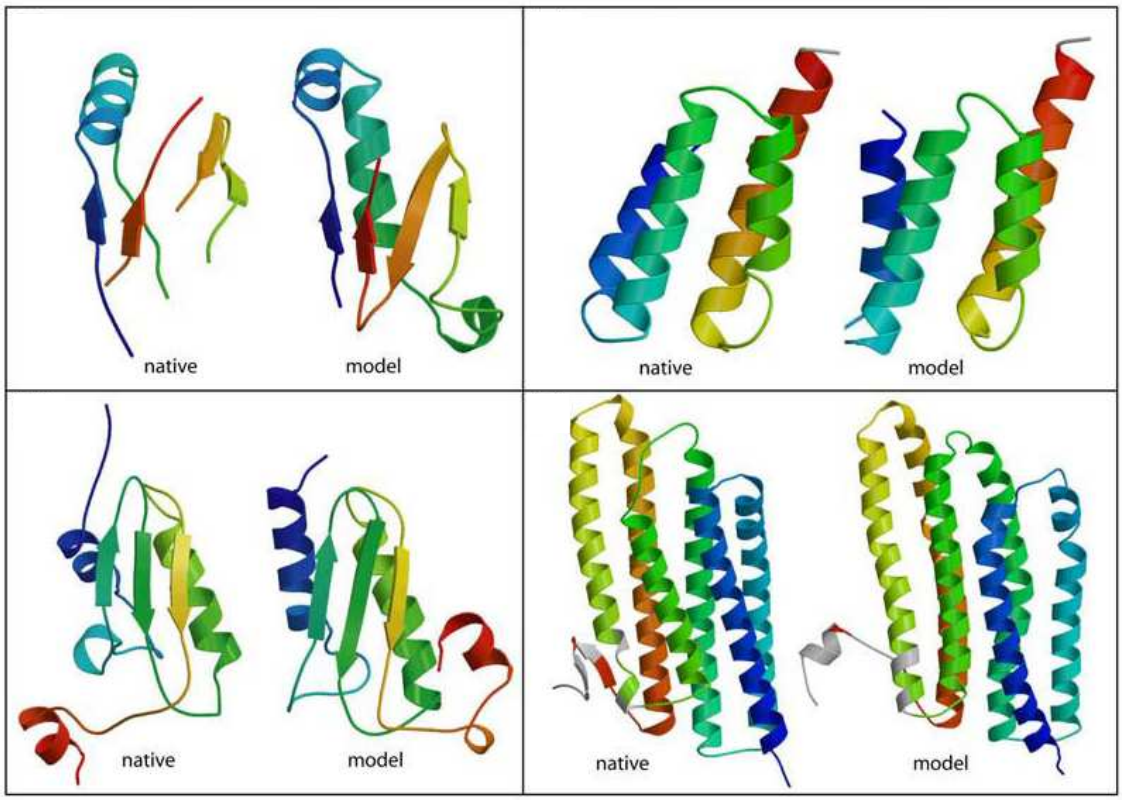


FIGURE 11 – Prédictions issues d'une session de CASP. *Prédictions de structures obtenues à l'aide du logiciel Rosetta [29] pour* CASP6*. Image originale en couverture du journal* PROTEINS : Structure, Function, and Bioinformatics*, volume 61 du 26 septembre 2005.*

Ces techniques s'appliquent aux protéines et aux ARN, avec plus de succès pour les protéines.

2.6 La structure quaternaire

2.6.1 Définition

La structure quaternaire est la géométrie de l'association de plusieurs sous-unités protéiques ou nucléiques. Certaines protéines ne sont fonctionnelles que sous forme d'oligomères. Il existe des oligomères formés de sous-unités identiques, comme par exemple le tétramère de la thymidylate synthase X, et des oligomères réunissant des sous-unités différentes, comme les histones. On parlera alors de complexes. Enfin, certaines protéines forment des polymères, constitués d'un très grand nombre de sous-unités, comme les polymères actine/myosine dans les muscles.

L'association de ces sous-unités est stabilisée par des interactions à courte distance, similaires à celles qui assurent la stabilité de la structure tertiaire (essentielle-

ment des liaisons hydrogène, des ponts salins et des interactions hydrophobes) [48, 125].

2.6.2 Détermination

L'existence d'oligomères de chaînes protéiques ou nucléiques, qu'elles soient ou non identiques, peut être déterminée par filtration sur gel ou centrifugation analytique par exemple. Mais leur existence peut aussi être déterminée par des méthodes de biochimie et de biologie moléculaire plus poussées et qui peuvent être utilisées de manière systématique, telles que, pour les protéines, l'analyse double-hybride [116], l'analyse par TAP-tag (ou FLAP-tag) couplée à la spectrométrie de masse [92, 109]. La géométrie de l'association peut être déterminée à basse résolution par diffusion des rayons X ou des neutrons aux petits angles, chromatographie sur gel ou encore par microscopie électronique à la fois pour les protéines ou les ARN.

La connaissance de l'interaction au niveau des acides aminés peut se faire, soit directement par la détermination de la structure par cristallographie aux rayons X, soit par l'étude des interactions par RMN, ou encore indirectement, par mutagenèse dirigée ou modification chimique sélective des chaînes latérales de certains acides aminés. Mais, en plus des contraintes associées aux deux méthodes vues précédemment, s'ajoutent les contraintes inhérentes aux complexes, telles que la taille, mais aussi, et surtout, leur instabilité. En effet, pour pouvoir être étudié d'un point de vue structural, un complexe doit être stable dans les conditions requises. Or, de très nombreux complexes sont transitoires. Ainsi, même s'il est désormais possible d'obtenir la structure de nombreuses protéines isolées de plus en plus rapidement, la résolution des structures de complexes reste difficile.

2.6.3 Prédiction

Le premier modèle de complexe protéine-protéine (trypsine/inhibiteur) a été réalisé en 1972 [20]. C'est en 1978 qu'est apparu le premier algorithme d'amarrage [259]. Les procédures d'amarrage utilisent les coordonnées atomiques des deux macromolécules partenaires, génèrent un grand nombre de conformations et leur attribuent un score [260]. Cette modélisation est en général assimilée à la recherche de modes d'association complémentaires entre deux molécules de forme prédéfinie. Un certain degré de flexibilité peut parfois être pris en compte, mais en général, l'amarrage protéine-protéine et protéine-ARN est principalement envisagé dans une approche d'association de corps rigides.

Ces méthodes s'appliquent à des protéines et acides nucléiques différents, mais peuvent aussi être envisagées pour déterminer l'état d'oligomérisation d'une protéine ou d'un ARN. Elles peuvent prendre en compte les symétries connues comme pour les protéines virales [12, 52, 199, 224], mais aussi utiliser des études plus fines des interfaces [19, 7, 200, 271].

3 Les complexes protéine-protéine et protéine-ARN

3.1 Fonctions

Au niveau moléculaire, la fonction d'une protéine ou d'un ARN est souvent subordonnée à l'interaction avec un certain nombre de partenaires. Les complexes interviennent à de nombreux niveaux et la compréhension de leur mécanisme de formation/association permet de mieux comprendre de nombreux processus. Pour se rendre compte de leur importance, on peut citer des assemblages tels que le ribosome, les anticorps/antigènes, les capsides virales ou encore les microtubules. Ainsi, la fonction d'une protéine ou d'un ARN ne peut être envisagée sans tenir compte des interactions.

3.2 Détection expérimentale biochimique protéine-protéine

Les interactions protéine-protéine sont présentes partout et en grand nombre. C'est la raison pour laquelle de nouvelles méthodes expérimentales d'analyse systématique sont développées [123]. Deux types sont présentés dans la suite, les méthodes d'analyse par double-hybride et celles utilisant des marqueurs.

3.2.1 Le double-hybride sur la levure

La première méthode utilisée pour étudier dans la levure les interactions protéine-protéine à grande échelle a été l'analyse par double hybride. Cette technique, mise au point en 1989, permet la détection indirecte de l'interaction, car celle-ci induit la formation d'un complexe moléculaire activant un gène rapporteur [81] (voir fig. 12). Cependant, dans cette détection, le nombre de faux positifs (les interactions détectées mais non présentes) et de faux négatifs (les interactions présentes non détectées) est très important. C'est donc une méthode relativement peu fiable, à moins de refaire un grand nombre de fois ces expériences, en plus d'expériences complémentaires, ce qui est relativement coûteux et long dans une approche génomique.

De plus, cette méthode ne peut détecter dans sa forme originelle que des complexes binaires. Or, la détection et la caractérisation de complexes multiprotéiques sont très importantes.

Deux études sur la levure utilisent le double-hybride pour la détection systématique [116, 248]. Il est toutefois très difficile de comparer ces études entre elles, en raison principalement des problèmes de fiabilité dus aux contraintes expérimentales.

3.2.2 Utilisation de marqueurs (*TAP-tag* et *FLAP-tag*)

Deux autres études ont été menées sur la levure *S. cerevisiae* [92, 109] pour identifier et comprendre le rôle de complexes cellulaires dans la cellule eucaryote. Des centaines de séquences codantes de levure ont été fusionnées à des cassettes d'ADN codant pour des marqueurs de purification. Puis, les souches de levure ont été cultivées, chacune exprimant une protéine cible marquée, et soumises à une procédure

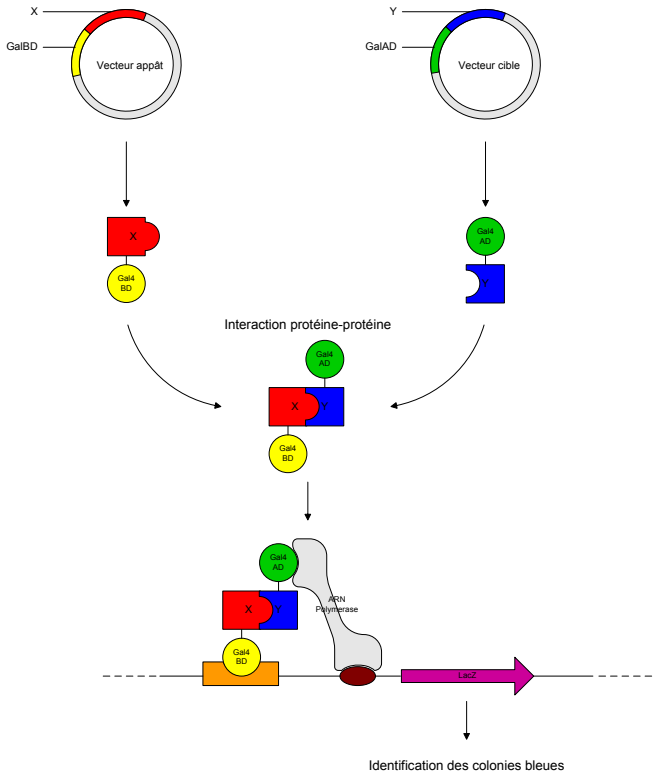


FIGURE 12 – Schéma de principe de détection des interactions protéine-protéine par double-hybride chez la levure. *La protéine Gal4 est l'activateur naturel des différents gènes intervenant dans le métabolisme du galactose. Elle agit en se fixant sur des séquences appelées UASG* (Upstream Activating Sequence GAL) *qui régulent la transcription. Les protéines étudiées (X et Y), partenaires potentiels d'interaction, sont fusionnées, l'une au domaine de fixation de Gal4 sur l'ADN (domaine DBD ou* DNA Binding Domain*), et l'autre au domaine de Gal4 activant la transcription (domaine AC ou* Activation Domain*). C'est ce qui donne à ce système le nom de double-hybride. Quand il y a interaction entre X et Y, les domaines DBD* (DNA-Binding Domain) *et AD* (Activation Domain) *sont associés et forment un activateur de transcription DBD-X/Y-AD. C'est cet activateur hybride qui va se lier à l'ADN au niveau des séquences qui contrôlent le gène rapporteur (les séquences UASG), permettant la transcription du gène par l'ARN polymérase II. Il suffit ensuite d'observer le produit du gène rapporteur, pour voir si un complexe s'est formé entre les protéines X et Y. Souvent, le gène LacZ est inséré dans l'ADN de la levure juste après le promoteur Gal4, de façon à ce que, si l'interaction a lieu, le gène LacZ, qui code pour la β-galactosidase, soit produit. Sur un substrat approprié, la β-galactosidase devient bleue, ce qui permet de déterminer simplement si l'interaction a lieu.* Image de J. Bernauer.

dans laquelle les complexes entiers, contenant la protéine marquée, ont été purifiés. Ensuite, les complexes ont été fractionnés par électrophorèse sur gel et leurs composants identifiés par spectrométrie de masse.

Il est très difficile de comparer les résultats obtenus par ces études car les jeux utilisés ne sont pas identiques et le protéome complet de la levure n'a pas pu être analysé. Globalement, ces études donnent des résultats en accord avec celles réalisées précédemment, mais dans le détail, les résultats et la complétude des données ne permettent pas de conclure.

De plus, il est aussi difficile, pour les mêmes raisons, de comparer les études utilisant des marqueurs et les études de double-hybride présentées précédemment.

L'ensemble de ces études expérimentales permet de prédire environ 15 000 complexes protéine-protéine potentiels pour le génome de la levure. Parmi ces 15 000 complexes, beaucoup s'avèreront être des faux positifs et il est certain qu'il existe également un grand nombre de faux négatifs.

3.3 Les méthodes d'amarrage

3.3.1 Le problème

Le but des méthodes d'amarrage est de prédire la structure d'un complexe à partir des structures ou modèles des partenaires isolés (voir fig. 13). Le problème se divise

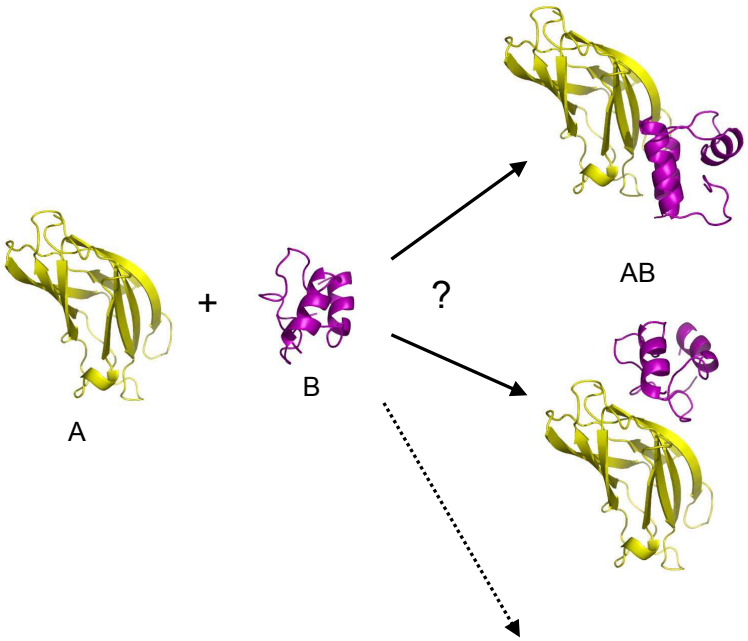


FIGURE 13 – Le problème de l'amarrage. Comment associer la protéine A et la protéine B ? Des configurations AB obtenues, laquelle est susceptible d'exister *in vivo* ? Image de J. Bernauer.

en deux étapes : d'abord, on explore l'espace pour obtenir toutes les conformations possibles et ensuite, on trie ces conformations en espérant classer en premier la conformation native observée expérimentalement.

Avec une approximation de corps rigides, si on considère chaque partenaire comme une sphère de 15 Å de rayon à la surface de laquelle les propriétés atomiques sont décrites sur une grille de 1 Å, une recherche systématique présente 10^9 modes distincts d'association [57]. La question est ensuite de déterminer, parmi ces modes d'association, lequel est le mode natif.

Pour pouvoir accéder aux changements de conformation et aux mouvements des chaînes latérales et des bases, le modèle doit être de type "soft", c'est-à-dire que les molécules doivent pouvoir légèrement s'interpénétrer et on doit considérer les molécules comme des ensembles de sphères articulées. Ainsi, il est possible de traiter aussi bien les molécules issues de résolution de structures de protéines seules, c'est-à-dire non-liées *(unbound)*, ou complexées, c'est-à-dire liées *(bound)*.

3.3.2 Les algorithmes

Le premier algorithme, inventé par Shoshana Wodak et Joël Janin [124, 259] à partir des travaux de Cyrus Levinthal [149] réalise une recherche de l'espace sur six degrés de liberté (cinq rotations et une translation) pour amener les deux molécules en contact une fois leur orientation fixée, et attribue un score simple en fonction de la surface de contact. Pour gagner du temps sur le calcul de la surface, une approximation à partir du modèle de Levitt [150] est réalisée. Cet algorithme a été amélioré en 1991 à l'aide d'une minimisation d'énergie [46].

D'autres types d'algorithmes, utilisant la complémentarité de surface, ont été mis en œuvre à partir d'une description en points critiques définis comme des "trous et bosses" *(knobs and holes)* [57, 145, 264]. Les solutions données correspondent à une concordance de groupes de quatre points critiques, laquelle est identifiée grâce à une triangulation de surface comme définie par M. Connolly en 1985 [56]. Cette méthode a été beaucoup améliorée en 1991 par H. Wang, avec la modélisation de la surface à l'aide d'une grille [255].

En 1992, un programme utilisant ces grilles pour les petites molécules a été modifié par I. Kuntz et ses collaborateurs, pour s'appliquer aux complexes protéine-protéine et a permis d'obtenir de bons résultats [175, 232] tout en générant de nombreux faux-positifs.

Des algorithmes de vision par ordinateur *(computer vision)* à partir de hachage géométrique ont ensuite étendu la méthode des "trous et bosses". En 1993 a été développé un algorithme qui fait correspondre des propriétés de surface à partir de triplets de points critiques qui sont stockés dans des tables de hachage [84, 163]. Cette méthode, très efficace pour les molécules de type lié, est très sensible aux faibles variations de surface, ce qui la rend rapidement inefficace pour les molécules de type non-lié.

3.3.3 La transformation de Fourier

Parmi toutes les méthodes de complémentarité de surface, celle utilisant la transformation de Fourier rapide (*Fast Fourier Transform* ou FFT), apparue dès 1991 [126], est l'une des plus simples et des plus utilisées [10, 35, 40, 41, 132, 143, 180, 235, 257]. Une grille cubique est tracée, et, à chaque point, on attribue un poids qui est négatif et important si le point est situé à l'intérieur de la protéine A, nul s'il est à l'extérieur et 1 s'il est proche de la surface ; on fait de même pour la protéine B.

Le produit est donc important et positif (donc défavorable) si les deux volumes moléculaires s'interpénètrent, et négatif (donc favorable) pour les points qui appartiennent à la surface d'une molécule et au volume de l'autre. Lorsque la molécule A est translatée par rapport à la molécule B, le score peut être rapidement calculé par transformation de Fourier rapide (FFT), si la grille de A est identique à la grille de B. La grille doit donc être redéfinie à chaque nouvelle orientation pour que la recherche soit complète.

Cette approche présente de nombreux avantages : les poids peuvent contenir des informations sur les propriétés physico-chimiques de la surface, et la résolution peut être ajustée en limitant le nombre de termes de Fourier calculés dans la somme.

Les résultats obtenus par cette méthode sont relativement bons dans une approche corps rigide [11, 39, 170], mais le temps de calcul associé est trop important pour une approche à grande échelle.

3.3.4 Algorithmes d'amarrage et partitionnement du problème

Ces quinze dernières années, en particulier grâce à l'expérience d'amarrage CAPRI *(Critical Assessment of PRediction of Interactions)* [118, 119, 120, 122, 261], plusieurs nouvelles méthodes ont vu le jour [234]. Cette expérience est un test à l'aveugle des algorithmes de docking de macromolécules qui doivent prédire le mode d'association de deux protéines à partir de leur structure tridimensionnelle. La structure du complexe, résolue expérimentalement, n'est dévoilée aux participants et publiée qu'à l'issue des soumissions.

Les nouvelles méthodes d'amarrage utilisent des techniques très variées telles que le hachage géométrique [83, 115, 188, 189, 190, 220, 225, 262], les algorithmes génétiques [91], les harmoniques sphériques [213], la dynamique moléculaire [32, 137, 238], la minimisation Monte-Carlo [100, 226], ou encore des méthodes de minimisation d'énergie ou de détection d'interfaces dirigées par des données biologiques [69, 178, 250] (voir section 2.1.1 page 11).

Le domaine de recherche a beaucoup progressé et l'une des conclusions de cette expérience est que l'on dispose d'algorithmes de recherche de complémentarité de surfaces performants [174]. Cependant, la deuxième étape du processus d'amarrage, à savoir le tri des configurations putatives obtenues par une fonction de score, reste à améliorer, car la seule méthode réellement performante à l'heure actuelle est l'expertise humaine. Les fonctions énergétiques classiquement utilisées ayant montré leurs limites [42, 53, 77, 79, 104, 108, 154], de nouvelles fonctions de score statistiques sont apparues. Essentiellement basées sur les propriétés physico-chimiques

des atomes, elles ont tout d'abord été utilisées pour le repliement et l'amarrage de petites molécules, puis adaptées à l'amarrage protéine-protéine [64, 265, 266, 267].

4 La tessellation de Voronoï et ses dérivées pour l'amarrage

La première utilisation connue de la tessellation de Voronoï est la modélisation de la répartition de l'épidémie de choléra de Londres par John Snow en 1854, dans laquelle est démontrée que la fontaine au centre de l'épidémie est celle de Broad Street, en plein coeur du quartier de Soho. Depuis lors, les applications utilisant cette construction sont nombreuses : en météorologie d'abord, par A.H. Thiessen, en cristallographie par F. Seitz et E. Wigner, qui ont aussi donné leur nom à cette construction ; mais aussi en physiologie (analyse de la répartition des capillaires dans les muscles), métallurgie (modélisation de la croissance des grains dans les films métalliques), robotique (recherche de chemin en présence d'obstacles) et bien d'autres.

La tessellation de Voronoï, ainsi que les autres tessellations qui en ont été dérivées (voir fig. 14), sont aussi beaucoup utilisées en biologie, où elles permettent de nombreuses représentations des structures des protéines [201].

Étant donné un ensemble de points appelés centroïdes, la *tessellation de Voronoï* divise l'espace au maximum en autant de régions qu'il y a de points (voir paragraphe 4.1.2 page 60). Chaque région, appelée cellule de Voronoï, est un polyèdre qui peut être considéré comme la zone d'influence du point autour duquel est tracée la cellule.

4.1 Constructions

Dans le cadre de l'analyse structurale des protéines, la tessellation de Voronoï a été utilisée pour la première fois par Richards en 1974 [211] pour évaluer, dans une protéine globulaire, les volumes des atomes, définis par les volumes de leurs polyèdres de Voronoï. Dans cette étude, Richards est le premier à proposer une solution à deux problèmes que l'on retrouve dans toutes les études qui utilisent cette construction. Tout d'abord, les atomes exposés au solvant ayant peu de voisins, leurs cellules de Voronoï sont grandes et ont un volume très grand, peu représentatif de leurs propriétés. Ensuite, cette construction considère tous les atomes comme équivalents, sans tenir compte de leur nature chimique.

Pour résoudre le premier problème, Richards a placé des molécules d'eau sur un réseau cubique entourant la protéine et a relaxé leurs positions. Cette méthode a été ensuite affinée par Gerstein et ses collaborateurs [94, 95, 243, 244, 245]. D'autres méthodes ont été proposées telles que :

- prendre en considération uniquement les atomes ayant une cellule de Voronoï de volume « raisonnable » [206] ;
- placer les molécules d'eau en utilisant la dynamique moléculaire [31, 37] ;
- utiliser une représentation d'union de sphères [172] ;

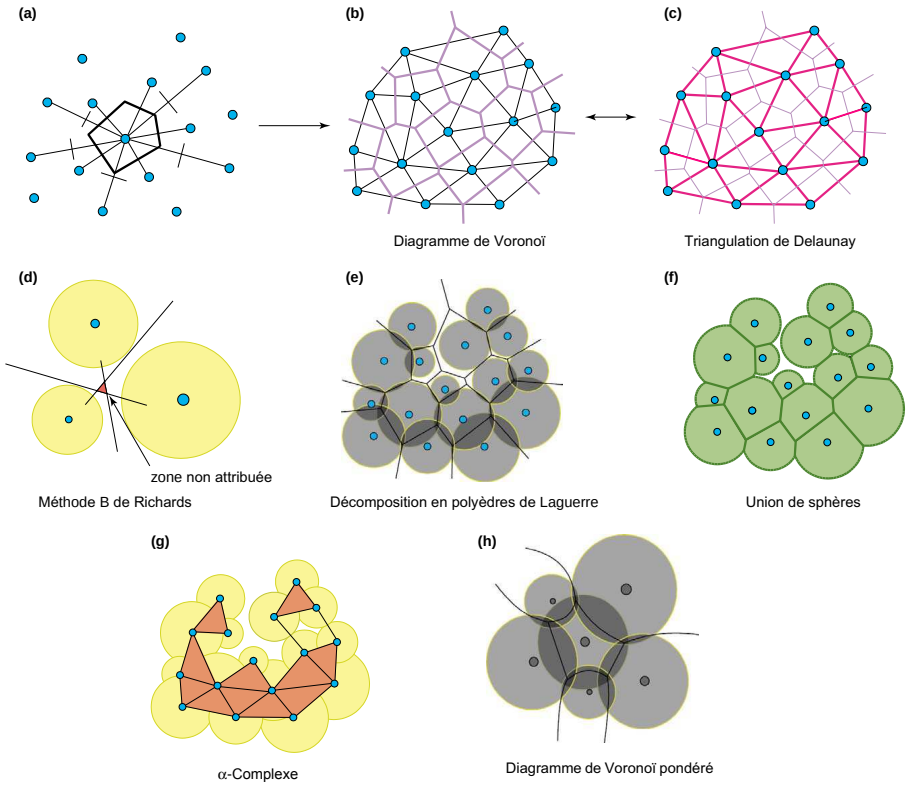


FIGURE 14 – Tessellation de Voronoï et constructions dérivées *(a) Construction d'une cellule de Voronoï : on trace la médiatrice entre un point donné et chacun des autres points et ensuite, on considère le plus petit polyèdre défini par ces médiatrices ; c'est la cellule de Voronoï de ce même point. (b) On obtient le diagramme de Voronoï (en violet) en répétant l'opération pour tous les points de l'ensemble. (c) La triangulation de Delaunay contient les arêtes roses et les triangles ainsi définis. C'est le dual du diagramme de Voronoï. (d) Dans la méthode de Richards, on ne considère pas la médiatrice, mais on définit une droite perpendiculaire au segment qui coupe celui-ci en fonction des poids attribués à chacun des atomes. Cela laisse une zone non attribuée. (e) Si on remplace les droites précédentes par les plans radicaux des sphères, on obtient à nouveau un pavage de l'espace : le diagramme de puissance ou tessellation de Laguerre. (f) L'intersection du diagramme de Laguerre et des sphères donne ce qu'on appelle l'union des sphères. (g) On définit une région restreinte comme une boule restreinte à sa région de Voronoï. L'α-complexe correspond alors aux arêtes et aux triangles définis par l'intersection de deux ou trois régions restreintes. L' α-shape est le domaine de l'α-complexe. (h) La surface de division d'un diagramme de Voronoï pondéré est définie par l'ensemble des points dont la distance aux deux points de référence est égale au rayon de la sphère correspondante plus une constante. Cette surface n'est pas plane, mais le diagramme correspondant est un pavage de l'espace.* Image de A. Poupon.

– utiliser un mélange entre la représentation en diagramme de puissance et la représentation en union de sphères (voir fig. 14).

Pour résoudre le problème des poids des atomes, Richards a proposé d'introduire des poids lors du placement des plans dans la construction de Voronoï. Cette méthode, appelée *méthode B de Richards*, a été très utilisée. Elle manque de rigueur mathématique car on trouve des volumes non attribués entre les cellules, l'intersection des plans n'étant plus réduite à un point. Cependant, Richards a montré que ce volume mort, bien que non nul, est petit en comparaison des volumes des atomes. Cette méthode a été de nombreuses fois améliorée [82, 212], jusqu'à utiliser le diagramme de Laguerre [93] ou le diagramme de Voronoï dit pondéré [58, 96], dans lequel les faces des cellules ne sont plus planes (voir fig. 14).

Une analyse formelle de toutes ces applications a été réalisée par Edelsbrunner et ses collaborateurs [72, 73, 74, 158, 159]. En plus des utilisations des tessellations de Voronoï/Delaunay/Laguerre, ils mettent en place la notion d'α-shape pour les protéines : c'est un sous-ensemble des segments issus de la tessellation de Delaunay qui sont contenus dans le volume de la protéine (voir fig. 14). Cela permet de modéliser l'intérieur de la protéine et de détecter les vides et les cavités [160].

4.2 Mesures

Toutes ces constructions ont permis de montrer que la tessellation de Voronoï est un bon modèle mathématique de la structure des protéines. Elle permet en particulier de montrer que les protéines sont des objets compacts, c'est-à-dire que la densité d'atomes à l'intérieur d'une protéine est comparable à celle observée dans les cristaux de petites molécules [95, 107]. De même, une analyse où les centroïdes sont les centres géométriques des acides aminés a permis de montrer que les protéines sont aussi des objets compacts au sens des modèles classiques des matières condensées en physique [3, 236].

Elle a également servi à l'analyse des cavités dans les structures [8, 72, 157, 198], à l'étude de propriétés mécaniques des protéines [136, 192, 222], à la mise en place de potentiels empiriques pour l'affinement de modèles structuraux [23, 34, 90, 140, 156, 161, 183, 256, 273], ou encore à la détection des hélices transmembranaires [1].

De telles méthodes ont aussi été utilisées pour détecter les cavités des protéines susceptibles d'interagir, mais aussi pour ajuster les ligands dans les poches ou encore étudier les interactions protéine-ADN [24, 28, 63, 186, 187]. Des études utilisant le modèle B de Richards ou la construction de Laguerre ont montré qu'à l'interface protéine-ADN et protéine-protéine, la densité de l'empilement est la même qu'à l'intérieur de la protéine pour la grande majorité des complexes [59].

Plus récemment, la tessellation de Voronoï a été employée pour la prédiction des complexes protéine-protéine, en particulier afin d'obtenir des descripteurs gros-grains pour la discrimination entre complexes cristallographiques et biologiques et pour les fonctions de score d'amarrage [27, 18, 17, 16, 15]. Ce type de méthode est détaillé au chapitre 4.

5 Fonctions de score

Traditionnellement, les fonctions de score pour la prédiction de la structure de macromolécules biologiques ont pour objectif de représenter l'énergie libre de la structure. Pour quantifier l'énergie libre d'une structure, les fonctions de score adoptent différentes méthodes. Parmi celles-ci, on compte des méthodes empiriques, inspirées des lois de la physique ou provenant de l'expertise des simulations de dynamique moléculaire. On compte aussi des méthodes basées sur la connaissance, *i.e.* issues de mesures sur des structures biologiques résolues expérimentalement.

De récents protocoles de prédiction des interactions protéine-protéine font état de l'usage de ces méthodes [18, 15, 22, 6, 27, 98]. RosettaDock [98] emploie un mélange de méthodes empiriques et de modélisation de lois physiques. Par exemple, deux partenaires en interaction ont au moins un certain nombre d'atomes en interaction : cette simple observation est modélisée par une simple fonction continue décroissante du nombre d'atomes en interaction. D'autres types de représentation des interactions locales, comme par exemple l'hydrophobicité, *i.e.* l'absence d'affinité entre les molécules de solvant et les groupements hydrophobes nécessitent des fonctions plus élaborées. Les fonctions de score développées pour et utilisées par RosettaDock [98] sont plus amplement détaillées dans la partie consacrée à l'évaluation des conformations par RosettaDock (voir section 2.1.1.2).

Des fonctions de score s'appuyant sur une modélisation simplifiée de la structure, dite gros-grain, ont déjà permis d'orienter les prédictions [18]. Ces fonctions de score sont généralement plus simples, moins coûteuses à calculer, à utiliser en amont d'une prédiction plus spécifique.

Pour prédire l'interaction, il est possible d'utiliser certaines données externes, *i.e.* ne provenant pas de la structure putative des molécules en interaction. On peut ainsi voir des fonctions de score utiliser la conservation de séquences entre deux protéines à travers l'évolution pour inférer le comportement à l'interaction d'une protéine par rapport à l'autre [22]. En effet, si une séquence est fortement conservée entre deux protéines, il y a de grandes chances pour que cette séquence joue un rôle dans au moins l'une des fonctions de chacune des deux protéines. Cependant, ces données externes ne sont pas forcément toujours accessibles, ce qui limite leur utilisation dans le cadre d'une prédiction d'interactions à grande échelle. Leur mauvaise interprétation peut aussi parfois être source d'erreur. La cible numéro 6 de CAPRI par exemple, bien que mettant en jeu des anticorps, traite d'une interaction n'impliquant pas le CDR *(Complementarity determining regions)* où a lieu l'interaction avec l'antigène. Il peut en outre s'avérer compliqué de comparer la qualité de prédictions effectuées par une méthode de prédiction incluant des données externes de façon optionnelle.

Mais aussi, des mesures plus complexes, comme un calcul de la complémentarité de forme entre les deux partenaires, ont permis de mieux résoudre des interactions de type clef-serrure. Il s'agit ici de l'utilisation de mesures faisant intervenir de façon plus importante la géométrie de la structure, sans forcément tenir compte de paramètres davantage d'ordre biophysique. La construction du diagramme de Voronoï a permis d'obtenir d'autres types de mesures géométriques sur la structure [15, 27]. De telles

mesures géométriques ont montré qu'il était possible de mieux évaluer l'empilement stérique des protéines à l'interaction, avec pour contrepartie de devoir construire le diagramme de Voronoï.

De manière plus générale, les fonctions de score jouent un rôle important en apprentissage automatisé dans la traduction d'un ensemble d'attributs caractérisant un exemple donné en une sortie numérique. Si cette sortie numérique peut parfaitement constituer l'estimation d'une observable, d'autres méthodes préfèrent l'utiliser pour trier, voire pour classer les exemples.

6 Apprentissage automatisé

Un panorama relativement exhaustif de l'état de l'art en apprentissage est disponible en référence [61]. Il propose une répartition des applications relevant du domaine de l'apprentissage artificiel selon deux grands axes : (i) reconnaissance des formes et (ii) extraction de connaissances à partir des données. Mais il est également possible de partitionner l'apprentissage automatique en fonction de la nature des données qui sont étudiées : *apprentissage supervisé*, où les données sont partiellement labellisées (étiquetées), *vs apprentissage non supervisé* (données sans labels).

Dans le cadre de l'analyse des protéines (prédiction d'interactions protéine-protéine et amarrage protéine-protéine), nous nous sommes focalisés sur des méthodologies relevant du domaine de l'apprentissage supervisé : apprentissage d'un *modèle prédictif* à partir des données connues.

6.1 Paradigme de l'apprentissage supervisé

Le paradigme de l'apprentissage supervisé peut se résumer de la façon suivante : *étant donné un ensemble d'exemples étiquetés, apprendre un modèle capable de prédire au mieux les étiquettes de nouveaux exemples*.

Soient $\mathcal{X}$ l'ensemble des *exemples* (ou données) et $\mathcal{Y}$ l'ensemble des *étiquettes* (notées aussi *classes*) pouvant être associées aux exemples. Dans le cadre des travaux présentés dans ce manuscrit, seules des étiquettes binaires ont été considérées : $\mathcal{Y} = \{+1, -1\}$ (notées également $\{+, -\}$ ou encore *presque-natifs* (+) et *leurres* (-) dans la suite du document).

Les données peuvent se répartir en deux catégories :

- les données déjà étiquetées, en général présentes en faible quantité car il est souvent très coûteux d'obtenir l'étiquette associée à une donnée (par exemple, obtenir la structure d'un complexe protéine-ARN par une expérience de cristallographie). Ces données seront utilisées pour apprendre un modèle permettant de prédire les étiquettes de nouveaux exemples.
- les données non étiquetées, qu'il est en général aisé d'obtenir. Dans notre cas, nous pouvons utiliser un algorithme de génération de conformations pour obtenir un large ensemble de conformations protéine-ARN non étiquetées.

L'ensemble des données déjà étiquetées est partitionné en un *ensemble d'apprentissage* et *ensemble de validation*. Nous désignerons l'ensemble d'apprentissage par $\mathcal{A} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ avec $x_i \in \mathbb{R}^d$ et $y_i \in \mathcal{Y}, \forall i \in \{1, \ldots, n\}$. x_i est un vecteur de dimension d où chaque dimension représente l'une des caractéristiques de l'exemple x_i.

L'apprentissage se déroule classiquement en trois phases, pour lesquelles on forme à partir de l'ensemble d'apprentissage un *jeu d'apprentissage* :

1. l'**apprentissage** sur le jeu d'apprentissage d'un modèle permettant de prédire au mieux les données d'apprentissage ;
2. l'**évaluation** de ce modèle sur des jeux de données extraits de l'ensemble d'apprentissage (par exemple grâce à une procédure de validation-croisée ou de *leave-one-out*) ;
3. le **test** du modèle obtenu sur un jeu de données étiqueté, qui est disjoint de l'ensemble d'apprentissage, l'ensemble de validation.

Le processus d'évaluation des performances d'un modèle appris nécessite d'utiliser des données non utilisées pour l'apprentissage afin de ne pas biaiser les évaluations de performances. Pour ce faire, l'évaluation s'effectue sur un modèle d'évaluation, spécifiquement appris pour la phase d'évaluation. Ce modèle d'évaluation est appris de la même manière que lors de la phase d'apprentissage, mais avec une partition de l'ensemble d'apprentissage : une partie des exemples est utilisée pour apprendre le modèle d'évaluation tandis que l'autre est utilisée pour l'évaluation proprement dite du modèle. Deux processus sont classiquement utilisés pour l'évaluation : la *validation-croisée* et le *leave-one-out*.

L'évaluation du modèle appris par **validation-croisée** consiste à partitionner les données de l'ensemble d'apprentissage en k parties disjointes, d'apprendre sur l'union de $k-1$ parties et d'évaluer ses performances sur la partie non utilisée. Ce processus est itéré k fois, ainsi tous les exemples de $\mathcal{A}$ auront été utilisés une fois en test et $k-1$ fois en apprentissage. Le choix de la valeur de k dépend de la taille des données. Les valeurs classiquement utilisées sont $k = 3$ ou $k = 10$.

L'évaluation par **leave-one-out** est une généralisation de la validation-croisée avec $k = n$. Ainsi, pour chaque exemple, un modèle est appris à partir de l'intégralité des données sauf l'exemple de test. Ce protocole d'évaluation est utilisé lorsque les données sont peu nombreuses et que le recours à la validation-croisée conduirait à se priver d'une trop grande partie des données pour l'apprentissage. Dès que les données sont trop volumineuses, le recours à cette méthode n'est plus viable car le coût de calcul devient rapidement prohibitif (apprentissage de n modèles).

Ces deux processus d'évaluation supposent une indépendance des exemples entre eux. Or, dans le cadre de l'amarrage protéine-ARN (ou protéine-protéine), les exemples ne sont pas indépendants. En effet, comme nous l'avons vu précédemment, nous disposons des structures de la protéine et de l'ARN, et à partir de ces deux structures, nous générons, via un algorithme d'amarrage, des candidats. Ces candidats vont représenter notre ensemble d'apprentissage. Un sous-échantillon de ces données sera étiqueté positif : la conformation dite native et les presques-natifs, si l'algorithme d'amarrage a réussi à en générer. Il existe donc un lien entre toutes les

conformations issues d'un couple protéine-ARN. Nous avons proposé une adaptation du processus *leave-one-out* pour prendre en considération ce lien. Il s'agit du procesus ***leave-"one-pdb"-out*** qui consiste à retirer non pas uniquement une conformation lors du processus d'apprentissage et d'évaluation, mais à retirer toutes les conformations associées à un couple protéine-ARN.

Les critères d'évaluation permettant de mesurer les performances des modèles appris sont essentiels dans tout processus d'apprentissage. De nombreux critères d'évaluation ont été proposés dans la littérature et nous présentons ci-après les critères les plus fréquemment utilisés et notamment ceux que nous manipulerons dans la suite de ce document.

6.2 Critères d'évaluation

Tout d'abord, lorsqu'un modèle prédictif est appliqué sur un jeu de données, nous pouvons mesurer, pour chaque étiquette, le nombre d'exemples correctement associés à cette étiquette, ainsi que le nombre d'exemples qui lui sont incorrectement associés. Ces informations sont rassemblées dans une matrice nommée la *matrice de confusion*.

Dans le cadre d'un modèle à deux classes, la matrice de confusion se représente classiquement sous la forme d'un tableau (voir tableau 1).

		Réel	
		+	-
Prédit	+	VP	FP
	-	FN	VN

TABLE 1 – Matrice de confusion, où VP représente le nombre de Vrais Positifs, FP le nombre de Faux Positifs, FN le nombre de Faux Négatifs et VN le nombre de Vrais Négatifs. Cette matrice de confusion restreinte à un problème à deux classes peut être étendue à un problème à *n* classes. La notion de Faux Positifs ou Faux Négatifs doit alors également être étendue.

6.2.1 Critères d'évaluation globaux

À partir de cette matrice de confusion, de nombreux critères d'évaluation peuvent être calculés. Parmi les plus utilisés, nous pouvons citer :

- la **précision** $P = \frac{VP}{VP+FP}$, qui représente le pourcentage de prédictions correctes associées à la classe positive (la même mesure peut être définie pour la classe négative) ;
- le **rappel** $R = \frac{VP}{VP+FN}$ qui représente le pourcentage d'exemples positifs étant correctement prédits positifs (de la même manière que pour la précision, il est possible de définir cette métrique pour la classe négative) ;
- le $F_{score}(\beta) = \frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R}$ qui permet d'agréger en une seule métrique la précision et le rappel ; Le paramètre β permet de pondérer la précision *vs* le rappel. Si $\beta < 1$, le poids de la précision devient plus important, inversement, lorque $\beta > 1$,

le poids de la précision diminue. Lorsque $\beta = 1$ la précision et le rappel ont la même importance. La valeur de β est très fréquemment fixée à 1 ;
- l'***accuracy*** $Acc = \frac{VP+VN}{VP+VN+FP+FN}$ qui permet d'évaluer la performance "globale" d'un modèle. Cette mesure représente le pourcentage de prédictions correctes toutes classes confondues ;
- la **sensibilité** $Se = \frac{VP}{VP+FN}$ qui est égale au rappel de la classe positive. Cette mesure est issue du domaine du traitement du signal et est largement utilisée dans le domaine médical ;
- la **spécificité** $Sp = \frac{VN}{FP+VN}$ qui correspond au rappel des négatifs. Cette mesure est également issue du domaine du traitement du signal. Son utilisation dans le domaine médical est toujours associée à la sensibilité. Ces deux mesures permettent d'évaluer l'efficacité d'un nouveau test médical en indiquant sa capacité à effectuer à la fois des prédictions correctes pour les positifs (sensibilité), tout en couvrant peu de négatifs (capacité évaluée par $1 - Sp$).

Toutes ces métriques fournissent une vision d'ensemble des performances d'un modèle en résumant en une unique valeur le comportement du modèle prédictif sur l'ensemble des données. D'autres métriques ou critères d'évaluation ont été proposés pour permettre d'obtenir une vision plus fine des performances d'un modèle. Des modèles donnant plus d'information qu'une variable binaire peuvent profiter de critères d'évaluation adaptés aux objectifs fixés. Nous parlerons par la suite de classifieurs pouvant soit donner une étiquette binaire aux exemples prédits soit leur attribuer un score permettant ainsi d'ordonner les exemples. Nous les appellerons par extension des *classifieurs*.

6.2.2 Critères d'évaluation "locaux"

Il est notamment devenu évident, depuis les années 2000, qu'il était insuffisant d'évaluer les performances d'un classifieur uniquement avec la précision et le rappel. De nouvelles métriques se sont rapidement imposées dans la communauté [165, 89], notamment des métriques permettent d'évaluer les classifieurs associant un score à chacune de leurs prédictions. Ce score, qui peut être assimilé à un degré de confiance dans la prédiction effectuée, permet alors d'ordonner les prédictions et ainsi d'obtenir plus d'informations qu'une étiquette.

La **courbe ROC** *(Receiver Operating Characteristic)* [252, 177] permet de visualiser le compromis entre la sensibilité et la spécificité. La courbe ROC associée à un classifieur idéal est constituée de deux segments : un premier segment reliant le point (0,0) au point (0,1) correspondant aux exemples positifs parfaitement ordonnés puis un second segment reliant le point (0,1) au point (1,1) correspondant aux exemples négatifs ayant tous des scores inférieurs aux scores des exemples positifs. Cette courbe représente un classifieur ayant la capacité de séparer parfaitement les positifs des négatifs (voir fig. 15).

Les courbes ROC permettent de visualiser rapidement les performances d'un ou plusieurs classifieurs. Afin de pouvoir comparer des classifieurs, notamment dans un cadre de recherche du meilleur classifieur (selon un ou plusieurs critères d'évaluation),

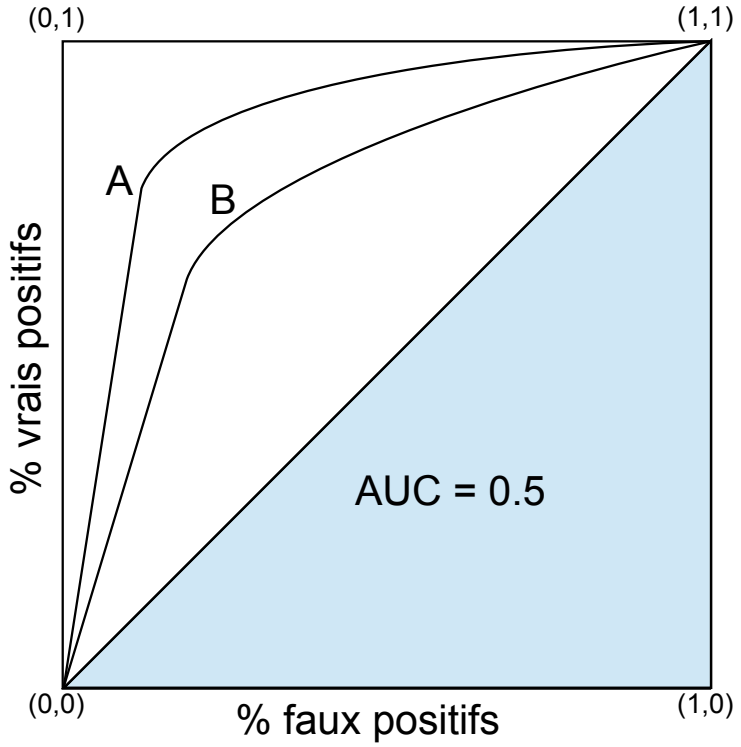


FIGURE 15 – Les courbes ROC associées aux classifieurs A et B permettent de visualiser la supériorité du classifieur A par rapport au classifieur B.

il est très utile de pouvoir comparer numériquement les performances de ces classifieurs.

L'**aire sous la courbe ROC** – que nous appellerons par la suite *ROC-AUC* – est très largement utilisée pour comparer les performances de plusieurs classifieurs. [165, 164] ont montré que l'aire sous la courbe ROC (AUC, *Area Under the Curve*) est une métrique plus fiable que l'*accuracy* pour comparer deux classifieurs.

De nombreux classifieurs "classiques" ont été adaptés pour pouvoir intégrer l'optimisation de la ROC-AUC dans leur critère d'apprentissage comme les SVM [208] ou les arbres de décision [80].

Sachant que, pour la problématique de l'amarrage protéine-ARN, seul un sous-ensemble très restreint de conformations candidates peuvent être proposées aux experts pour une validation expérimentale, il est nécessaire de se focaliser sur des métriques permettant d'identifier un sous-ensemble de conformations intéressantes.

L'ensemble des critères d'évaluation locaux utilisés sera présenté en détails dans la section 1.4.

Chapitre 1

Données

1.1 Jeux de données de complexes protéine-ARN

Dans cette étude, quatre jeux de données provenant de la littérature sont utilisés : la *Protein-RNA Interface DataBase* (PRIDB) comprenant deux versions RB1179 et RB199 [152, 254], le *Benchmark* I protéine-ARN [9] et le *Benchmark* II protéine-ARN [204] (voir tableau 1.1).

1.1.1 Jeu de référence des complexes protéine-ARN connus

La *Protein-RNA Interface DataBase* (PRIDB)[4] se décline en deux versions : la PRIDB redondante dénommée RB1179 [152] et la PRIDB non redondante RB199 [254]. Elle est disponible en téléchargement sur le site de la *Iowa State University*[4]. Ce jeu de données regroupe l'ensemble des fichiers des complexes protéine-ARN de la *Protein Data Bank* (PDB) extraits de façon semi-automatique contenant au moins une surface d'interaction entre une protéine et un ARN. Sa version redondante (RB1179) contient 1 170 complexes protéine-ARN. Sa version non redondante (RB199) tire son nom des 199 chaînes d'acides aminés qu'elle contient extraites de la PDB en mai 2010.

L'extraction des données de la PRIDB non redondante est faite en respectant les critères suivants :

- s'assurer de la qualité des données, en extrayant uniquement les structures résolues par cristallographie et ayant une résolution d'au plus 3.5 Å ;
- les structures extraites doivent contenir une chaîne d'au moins 40 acides aminés, dont au moins 5 en interaction avec un ARN d'au moins 5 acides nucléiques ;
- un acide aminé est considéré en interaction avec un ARN si l'un de ses atomes est à au plus 5 Å d'un atome de l'un des acides nucléiques ;
- pour s'assurer de la non redondance entre les données, l'identité de séquence maximale entre chacune de ses chaînes d'acides aminés est d'au plus 30 %.

La PRIDB non redondante RB199 contient 133 complexes. Or, plusieurs de ses complexes mettent en jeu plus de 2 partenaires. Nous n'utilisons donc que 120 des complexes de la PRIDB (voir section 1.2).

4. http ://pridb.gdcb.iastate.edu/download.php

1.1.2 Jeux d'évaluation des procédures d'amarrage : *Benchmarks*

Le *Benchmark* I [9] protéine-ARN contient 45 complexes protéine-ARN tandis que le *Benchmark* II protéine-ARN contient 106 complexes.

Tous les complexes du *Benchmark* I ont une version liée et non liée de la protéine (voir section 2.1.1). 11 complexes ont une version liée et non-liée (ou modélisée) de l'ARN. Par la suite, on utilise l'ensemble des 45 complexes de ce jeu. Parmi ces 45 complexes, seuls 11 complexes ne sont pas dans les 120 complexes de la PRIDB non redondante.

Sur les 106 complexes du *Benchmark* II, 76 complexes ont une version non liée de la protéine. Par la suite, on utilise aussi ces 76 complexes pour l'évaluation. Parmi ces 76 complexes, 36 complexes protéine-ARN ne sont pas dans la PRIDB non redondante. Toutefois, il y a un complexe du *Benchmark* II (1eiy) qui met en jeu les deux mêmes partenaires qu'un complexe présent dans la PRIDB (2iy5). Seuls 35 complexes du *Benchmark* sont donc utilisés, 1eiy étant écarté et 2iy5 étant utilisé dans la PRIDB. Il y a 5 complexes identiques entre les 45 complexes du *Benchmark* I et les 36 complexes du *Benchmark* II absents de la PRIDB non redondante (voir fig. 1.1). Il y a aussi 1 complexe du *Benchmark* I (2drb) mettant en jeu les deux mêmes partenaires qu'un complexe du *Benchmark* II (2dra), même si le code pdb du complexe est différent. C'est 2dra qui est utilisé.

Dans l'ensemble des *Benchmarks*, nous avons donc un total de 40 complexes (5 du *Benchmark* I et 29 du *Benchmark* II et 6 de leur union) différents des complexes déjà présents dans la PRIDB non redondante. Pour éviter la redondance, ce sont ces 40 complexes que nous utilisons concernant les *Benchmarks*.

1.2 Nettoyage des données

Comme le protocole d'amarrage utilisé prend en entrée deux partenaires et que nous cherchons tout d'abord à mettre en place une procédure de prédiction pour les complexes binaires (le comportement des assemblages multiples étant bien plus difficile à décrire), nous n'utilisons que des complexes avec deux partenaires en interaction. Les complexes de RB199 avec plus de deux partenaires en interaction (1a34 et 2q66) sont donc retirés. Pour des raisons de complexité de calcul et des interactions en présence, les complexes de RB199 mettant en jeu le ribosome complet [5] sont aussi retirés. Après ces filtres, le jeu de données issu de RB199 contient 120 complexes (voir tableau S10).

Les atomes d'hydrogène ne sont souvent pas présents dans les structures 3D contenues dans la PDB. Aussi, les paramètres des fonctions de score ne sont définis que pour les atomes lourds des acides aminés (resp. acides nucléiques) standards. C'est pourquoi, pour l'ensemble des complexes utilisés, les hydrogènes sont retirés et les acides aminés (resp. acides nucléiques) non standards transformés en acides aminés (resp. acides nucléiques) standards. Ce remplacement des acides aminés

5. Liste des complexes de RB199 mettant en jeu le ribosome complet : 1hr0, 1vqo, 2j01, 2qbe, 2vqe, 2zjr, 3f1e, 3huw, 3i1m, 3i1n, 3kiq

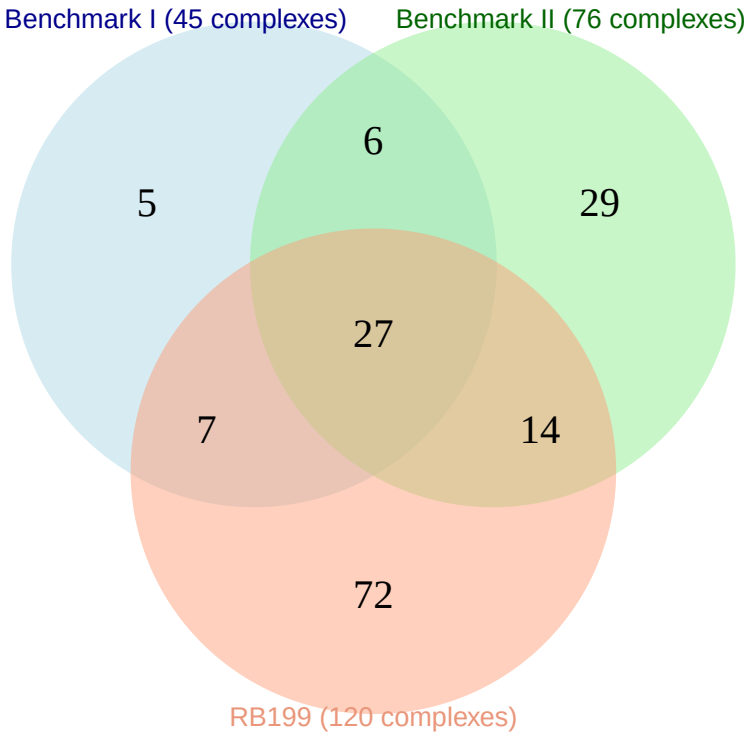


FIGURE 1.1 – Diagramme de Venn du nombre de complexes protéine-ARN utilisés par jeu de données externe, extrait de la PDB : la PRIDB non redondante munie de 120 complexes, le *Benchmark* I avec 45 complexes et le *Benchmark* II comportant 76 complexes avec une version non liée de la protéine.

(resp. acides nucléiques) non standards se fait en utilisant la correspondance de la PDBeChem [6], qui recense les dénominations chimiques de la PDB. Ainsi pour chaque acide aminé (resp. acide nucléique) non standard, on peut obtenir l'acide aminé (resp. acide nucléique) standard le plus proche pouvant le substituer. Cependant, une telle substitution peut avoir un impact sur le résultat biologique et ainsi rendre la structure considérée comme native peu fiable.

1.3 Utilisation des données

Les données doivent être utilisées pour définir et étiqueter les exemples. Il faut de plus regrouper les exemples en jeux d'apprentissage et de test pour la modélisation de la fonction de score comme pour l'évaluation du modèle. Mais le choix des étiquettes dépend du contexte biologique et nécessite donc de s'y attarder. C'est ce que nous verrons en section 1.3.1. Nous verrons ensuite comment obtenir les exemples en section 1.3.2, leur étiquetage en section 1.3.4 et leur regroupement en différents jeux en section 1.3.5.

6. http ://www.ebi.ac.uk/pdbe-srv/pdbechem/

1.3.1 Contexte biologique : définitions

Dans une expérience de prédiction de structure de complexe par *docking in silico*, nous obtenons un grand nombre de structures potentiellement proches de la structure biologique qui n'est pas connue (voir section 2.1.1). Dans la suite, nous souhaitons extraire des structures de complexes biologiques vraisemblables dans un ensemble de structures issues d'une expérience d'amarrage *(docking)*. Dans cet objectif, nous mettons en œuvre une approche d'apprentissage supervisé : il nous faut donc étiqueter les données en fonction de leur intérêt biologique sur des exemples pour lesquels on connait déjà la solution.

Ainsi :

- les complexes de la PDB sont les structures obtenues expérimentalement appelées *natifs* ou dites natives ;
- les complexes *candidats* sont ceux pour lesquels on cherche à déterminer s'ils sont biologiquement *natifs* ;
- les complexes *leurres* sont les complexes non-natifs, *i.e.* ceux qui ne sont vraisemblablement jamais formés dans la cellule ;
- les complexes proches de la solution native (selon un seuil à déterminer en fonction du but recherché) sont les *presque-natifs*.

Les jeux de données de référence sont constitués de structures 3D représentant la solution biologique de l'interaction. Ce sont donc des *natives*. La PRIDB en contient 120, les *Benchmarks* 40 (11 issus du *Benchmark* I et 29 issus du *Benchmark* II).

Les 120 structures natives sont utilisées pour générer un *ensemble de perturbations*. Cet ensemble de perturbations contient l'ensemble des exemples utilisés pour l'apprentissage et l'évaluation. Il correspond donc à l'ensemble d'apprentissage (voir section 0.6.1).

Comme vu dans le chapitre précédent, pour correctement valider le modèle de prédiction, il est nécessaire d'utiliser des données différentes des données utilisées pour l'apprentissage. Une validation du modèle est donc effectuée grâce à un *jeu de validation* généré de la même manière mais à partir des 40 structures des *Benchmarks*. Ce jeu de validation est utilisé pour évaluer la fonction de score générée à partir de l'ensemble de perturbations.

1.3.2 Ensemble des perturbations pour l'apprentissage

Comme cela a été fait dans la littérature pour l'amarrage protéine-protéine [98], des exemples positifs et négatifs sont générés par perturbation des structures natives. La perturbation des structures natives est un procédé générant un échantillon gaussien des candidats autour de la structure native. Pour chaque complexe, 10 000 candidats sont générés par perturbation des coordonnées des atomes de l'ARN par rapport aux coordonnées des atomes de la protéine. L'opération de perturbation consiste en 1 translation et 3 rotations pour modifier les coordonnées des atomes de l'ARN. Les amplitudes de la translation et des rotations sont chacune choisies selon une loi gaussienne de variance 1 et de moyenne variable. La moyenne des amplitudes de la translation et des rotations est adaptée en fonction du complexe. La moyenne

est déterminée de telle sorte que, pour un complexe donné, il existe au moins 30 candidats suffisamment proches de la structure native et 30 candidats suffisamment éloignés de la structure native (voir section 1.3.3). Pour satisfaire ces deux conditions sur les candidats obtenus, 3 jeux distincts sont utilisés :

- la perturbation standard a un jeu de moyennes de valeurs 3 Å pour la translation et 8 ° pour les rotations ;
- la perturbation restreinte a un jeu de moyennes de valeurs 1 Å et 4 °;
- la perturbation étendue a un jeu de moyennes de valeurs 9 Å et 27 °.

1.3.3 Mesure de similarité : le RMSD

Le RMSD *(Root-Mean-Square Deviation)* est une mesure utilisée pour évaluer la dissimilarité entre deux structures représentant des conformations différentes d'un même complexe. Dans une telle comparaison, les deux jeux d'atomes sont identiques ou comparables, mais avec des coordonnées spatiales différentes. Le RMSD utilise la distance entre chaque atome dans les deux structures, une fois que les structures sont alignées. Le RMSD est calculé comme la distance moyenne entre les atomes de ces deux structures. Considérons deux structures s_1 et s_2 d'un même complexe de N atomes, soit d_i la distance entre l'atome i de s_1 et l'atome i de s_2 (voir éq. 1.1) :

$$\mathrm{RMSD}(s_1, s_2) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{1.1}$$

Le RMSD est couramment utilisé sous diverses versions pour évaluer la dissimilarité entre une structure native et un candidat. Parmi ces versions du RMSD, il y a le LRMSD et le IRMSD utilisés par CAPRI, calculés sur une restriction des atomes : uniquement sur le ligand pour le LRMSD, uniquement sur l'interface pour le IRMSD [147] (voir fig. 1.2). L'alignement diffère selon le RMSD calculé. Pour le LRMSD, l'alignement s'effectue uniquement sur les atomes du squelette des acides aminés de la protéine. Pour le IRMSD, ce sont les atomes à l'interface qui sont alignés. Le RMSD peut aussi être calculé soit sur l'ensemble des atomes de chacun des acides aminés et acides nucléiques considérés, soit uniquement à partir d'un atome de référence par acide aminé ou acide nucléique. C'est ici le carbone α qui est utilisé comme atome de référence de l'acide aminé. Pour l'acide nucléique, c'est l'atome de phosphore qui est utilisé comme atome de référence.

Cette mesure de similarité est utilisée dans la génération des candidats. Les deux conditions fixées sur le nombre minimum de candidats à générer sont :

- un IRMSD $\leq$ 5 Å pour les 30 candidats devant être suffisamment proches de la structure native ;
- un IRMSD $>$ 8 Å pour les 30 candidats devant être suffisamment éloignés de la structure native.

Ces deux conditions ont pour objectif de conserver une distribution relativement homogène de chaque nuage de candidats. Il reste ensuite à comparer les candidats générés entre eux pour les étiqueter.

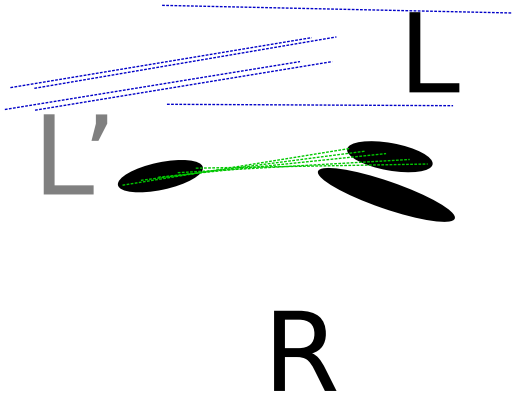


FIGURE 1.2 – Représentation schématique en gris des acides aminés et acides nucléiques sélectionnés pour le calcul du LRMSD et du IRMSD. La protéine (ou récepteur) est dénommée R. L'ARN (ou ligand) est appelé L et L' correspond à la position de L dans une seconde structure de l'interaction entre R et L. Le LRMSD est calculé sur les atomes du ligand L, les atomes du récepteur R étant alignés, à partir des distances en vert et en bleu (seules quelques distances sont affichées). Le IRMSD est calculé sur les atomes des acides aminés et acides nucléiques à l'interface, en noir, à partir des distances en vert, les atomes à l'interface étant alignés.

1.3.4 Étiquetage

Une fois les jeux de données générés, il faut choisir la version de RMSD utilisée et le seuil en RMSD en-dessous duquel un candidat est considéré comme un presque-natif. La version de RMSD impacte les possibilités de comparaison des structures candidates de différents complexes. Une version de RMSD très sensible à la taille des partenaires (en nombre d'acides aminés et d'acides nucléiques) donne une signification très différente au seuil fixe pour des complexes de tailles différentes. Un complexe de petite taille doit être bien plus éloigné de la structure native qu'un complexe de grande taille pour arriver au seuil fixé. Comme nous choisissons un seuil fixe, indépendant de la taille des complexes, il est préférable d'avoir une version de RMSD peu sensible à la taille des complexes. Nous choisissons le IRMSD qui, malgré sa dépendance à la taille de l'interface, reste la version CAPRI de RMSD le moins sensible à la taille de chacun des deux partenaires.

Le seuil de IRMSD a un impact sur la tolérance que le modèle de prédiction a sur la divergence entre un candidat considéré presque-natif et la structure native. Un seuil élevé accepte comme presque-natif des structures candidates très éloignées de la structure native, donnant une prédiction avec peu de signification. Certes, un tel modèle peut plus facilement prédire si un candidat est presque-natif, mais un candi-

dat prédit presque-natif par ce biais peut être très proche comme très éloigné de la structure native. Un seuil bas génère trop peu de presque-natifs, ce qui limite d'autant le nombre de candidats disponibles pour l'apprentissage. Pire, un seuil trop bas ne permettrait pas d'obtenir un modèle de fonction de score utilisable pour affiner une structure 3D. Nous choisissons un seuil en IRMSD de 5 Å :

- les candidats de IRMSD $\leq$ 5 Å sont considérés comme des *presque-natifs* ;
- les candidats de IRMSD $>$ 5 Å sont considérés comme des *leurres*.

Le seuil en IRMSD de 5 Å n'est pas choisi au hasard. Pour un complexe protéine-protéine, le critère CAPRI pour une solution acceptable est d'avoir un IRMSD $<$ 4 Å. Comme nous évaluons des complexes protéine-ARN, où l'ARN est une molécule plus flexible, nous admettons une marge d'erreur légèrement supérieure, pour un seuil en IRMSD de 5 Å.

Jeu de données	Nb. de structures	Provenance	Utilisation
PRIDB redondante RB1179	1 170	PDB	Mesures
PRIDB non redondante RB199	120	PDB	Génération
Ensemble de perturbations	1 200 000	Généré	Apprentissage
Benchmark protéine-ARN I	45 (11)	PDB	Génération
Benchmark protéine-ARN II	76 (29)	PDB	Génération
Jeu de validation	400 000	Généré	Validation

TABLE 1.1 – Nombre de structures 3D présentes, provenance et utilisation pour chaque jeu de données. Pour les *Benchmarks* I et II, le nombre de structures 3D entre parenthèses correspond au nombre de structures 3D n'étant pas déjà présentes dans la PRIDB non redondante RB199. Il s'agit aussi du nombre de structures 3D effectivement utilisées dans la génération de candidats. Chaque jeu de données généré est issu des jeux de données extraits de la PDB dans la même section, qui sont indiqués comme utilisés pour la génération de candidats. Les mesures effectuées sur la PRIDB redondante sont des mesures statistiques.

1.3.5 Constitution des jeux d'apprentissage et de test

Les jeux d'apprentissage sont constitués par échantillonnage sans remise des candidats des deux classes pour chaque structure native :

- 30 presque-natifs ;
- 30 leurres de IRMSD $>$ 8 Å.

Pour que cet échantillonnage soit possible, pour chaque structure native, l'étape de génération des candidats a assuré qu'au moins 30 presque-natifs et 30 leurres de IRMSD $>$ 8 Å soient générés. Le seuil en IRMSD de 8 Å permet de s'assurer qu'il existe au moins 30 leurres suffisamment éloignés des presque-natifs.

Les jeux de test contiennent l'ensemble des candidats générés. Le jeu de test de la fonction de score finale est le jeu de validation, constitué des candidats générés grâce

aux 40 complexes issus des *Benchmarks* I et II. Le faible nombre d'interactions non redondantes disponibles incite à utiliser au maximum les données disponibles.

Pour effectuer une première évaluation avant de générer la fonction de score à partir des candidats issus des 120 structures natives, une stratégie adaptée du *leave-one-out* est employée : le *leave-"one-pdb"-out* (présenté au chapitre précédent, section 0.6.1). Avec le *leave-"one-pdb"-out*, pour chaque structure native, une fonction de score est apprise, avec :

- en jeu de test, les 10 000 candidats de la structure native ;
- en jeu d'apprentissage, les 30 presque-natifs et 30 leurres des 119 autres structures natives.

Cet ensemble de différents jeux de données nous permet, tout en utilisant au maximum la quantité relativement faible de données disponibles, de valider le protocole d'apprentissage à chaque étape.

1.4 Mesures d'évaluation locales

Les données issues de l'évaluation des différents jeux de données nécessitent l'utilisation de mesures d'évaluation adaptées pour comparer les méthodes et leurs mises en œuvre sur chaque jeu de données. Les mesures d'évaluation globales ont déjà été détaillées précédemment (voir section 0.6.2).

Les mesures d'évaluation locales sont plus spécifiques à l'objectif qui nous intéresse de discriminer les presque-natifs des leurres dans une prédiction d'interactions protéine-ARN.

Parmi celles-ci, le **nombre de presque-natifs dans le top10** est sans doute le plus important, puisqu'il s'agit de compter le nombre de presque-natifs contenus dans les 10 premiers candidats lorsque les candidats sont triés selon le score. Or, le nombre de candidats que CAPRI accepte de recevoir comme solutions envisagées par une fonction de score est justement de 10. Évidemment, ce nombre va de 0 à 10 et, plus ce nombre est élevé, meilleure est la prédiction. Pour qu'une fonction de score soit considérée comme ayant réussi à modéliser l'interaction, elle doit avoir au moins 1 presque-natif dans le top10.

Les **diagrammes d'énergie en fonction du RMSD** permettent d'identifier les entonnoirs *(funnels)* de la prédiction. Sur le diagramme d'énergie en fonction du RMSD, il s'agit d'identifier une forme en entonnoir des points. Cet entonnoir se situe idéalement la pointe en bas à gauche du graphique, sa partie évasée partant en diagonale à droite (voir fig. 1.3). Un entonnoir implique que la fonction de score indique, pour des candidats de RMSD donnée, une plage de valeurs de score inférieure pour des candidats de RMSD supérieure. La découverte d'un entonnoir est le signe que la fonction de score évaluée est utilisable pour affiner la structure 3D de l'interaction recherchée. Les candidats se trouvant dans le coin supérieur gauche sont des presque-natifs mal prédits. De la même manière, les candidats se trouvant dans le coin inférieur droit sont des leurres mal prédits. Les candidats se trouvant dans le coin inférieur gauche sont les presque-natifs correctement prédits et ceux dans le coin supérieur droit des leurres correctement prédits. Les candidats du top10 sont les 10 premiers en partant du bas

du diagramme. C'est le diagramme de l'énergie en fonction du IRMSD qui est utilisé comme critère d'évaluation (noté *EvsRMS*).

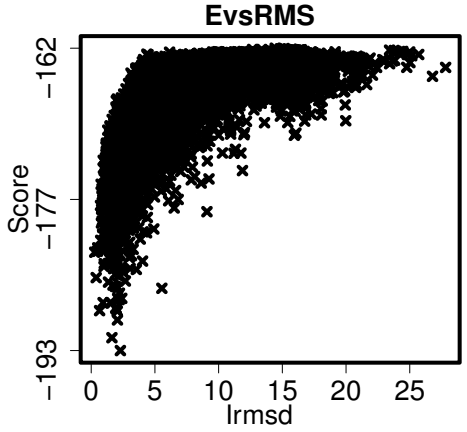


FIGURE 1.3 – Exemples de diagramme d'EvsRMS : le diagramme d'énergie en fonction du IRMSD permet la détection d'entonnoir. Voici un exemple d'entonnoir, avec la majorité des candidats suivant un entonnoir partant du coin inférieur gauche pour s'évaser dans le coin supérieur droit. Une telle courbe permet de montrer que la fonction de score proposée est utilisable pour de l'affinement de structure, une fois l'épitope de l'interaction identifié.

Soit C un ensemble de candidats généré à partir d'un même complexe c, trié en énergie selon une fonction de score. Le **score d'enrichissement** $ES(C)$ mesure la performance d'une fonction de score pour maximiser la proportion des 10 % premiers candidats triés en IRMSD parmi les 10 % premiers candidats triés en énergie. Ainsi, le score d'enrichissement est le nombre de candidats de C dans les 10 % premiers candidats à la fois en énergie et en IRMSD, divisé par le nombre total de candidats de C multiplié par 100 (voir éq. 1.2). En reprenant l'exemple du diagramme d'EvsRMS, les candidats à la fois dans les 10 % premiers candidats en score et les 10 % premiers candidats en IRMSD correspondent aux candidats dans le coin inférieur gauche du graphique, si l'on sépare le graphique verticalement et horizontalement à 10% des candidats en IRMSD et 10% des candidats en énergie (voir fig. 1.4).

Plus le score d'enrichissement est élevé, plus le tri est enrichi, c'est-à-dire plus la proportion de presque-natifs est importante dans les 10 % premiers candidats du tri. Le score d'enrichissement va de 0 à 10. Dans le cas d'un tri parfait, le score d'enrichissement vaut 10. Quand il y a indépendance entre tri en énergie et tri en IRMSD, le score s'enrichissement vaut 1. Un score d'enrichissement inférieur à 1 indique qu'aucun enrichissement n'est observé, correspondant à une mauvaise fonction de tri.

$$ES(C) = 100\frac{\#\left(top10\%_{\text{IRMSD}}(C) \cap top10\%_{\text{ENERGY}}(C)\right)}{\#(C)} \tag{1.2}$$

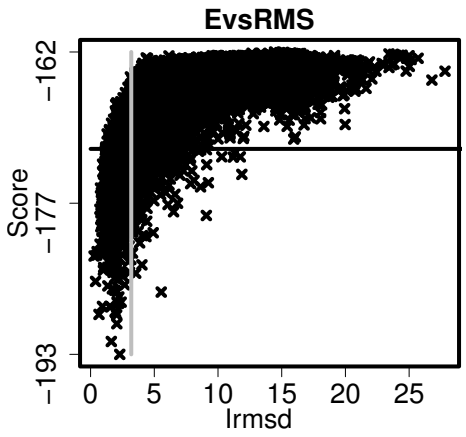


FIGURE 1.4 – Illustration du score d'enrichissement sur une courbe EvsRMS : un trait horizontal (resp. vertical) divise le graphique en deux parties de telle sorte que 10 % des candidats soient toujours en-dessous (resp. à gauche) de la séparation. Les candidats dans le cadran inférieur gauche sont les presque-natifs bien prédits, contrairement aux candidats dans le cadran supérieur gauche et le cadran inférieur droit. Les leurres correctement prédits se trouvent dans le cadran supérieur droit. Le score d'enrichissement équivaut à la proportion de candidats dans le cadran inférieur gauche parmi l'union des candidats en-dessous de la ligne de séparation horizontale et des candidats à gauche de la ligne de séparation verticale.

Chapitre 2

Approche Rosetta et adaptation

2.1 Présentation de RosettaDock

RosettaDock est un outil de prédiction d'interactions entre macromolécules biologiques. La prédiction d'interactions de RosettaDock est conçue et optimisée pour modéliser les interactions entre protéines. RosettaDock s'appuie sur un protocole appelé *amarrage (docking)*. D'autres protocoles en lien avec l'amarrage sont aussi disponibles via RosettaDock. Nous traiterons d'abord de l'amarrage avant de continuer sur les autres protocoles.

2.1.1 Amarrage

L'amarrage consiste à prédire la structure 3D représentant l'interaction de deux partenaires donnés pour chacun desquels la structure native est connue. Classiquement, la prédiction de la structure 3D de l'interaction est réalisée en deux grandes étapes :

1. la génération d'un large ensemble de candidats (de conformation plausible pour l'interaction des deux partenaires) ;
2. puis le tri de l'ensemble des candidats selon différents critères permettant de rendre compte de la qualité des candidats pour représenter l'interaction (taille de l'interface, nombre de résidus en contact *etc.*).

Dans les deux prochaines sections, nous allons séquentiellement détailler chacune de ces deux étapes.

2.1.1.1 Génération des candidats

L'étape de génération des candidats consiste à construire plusieurs amarrages 3D possibles des deux partenaires étudiés. Un candidat généré est défini par les coordonnées spatiales des atomes de ses partenaires. Afin de réduire le temps de calcul de la génération, seules les coordonnées d'un des deux partenaires sont modifiées, pour déplacer ce partenaire autour de l'autre partenaire, qui reste immobile. En règle

générale, le partenaire de plus petite taille est considéré comme mobile, alors que le plus volumineux (en nombre d'acides aminés ou d'acides nucléiques) est fixe. Dans notre cadre d'étude, le partenaire mobile est l'ARN et le partenaire fixe est la protéine, indépendamment de leurs tailles respectives.

L'espace des candidats possibles d'une interaction a 6 degrés de liberté (voir fig. 2.1) :

- la translation ρ suivant l'axe X ;
- la rotation χ suivant l'axe X ;
- la rotation ϕ_1 suivant l'axe Y_1 ;
- la rotation θ_1 suivant l'axe $Z1$;
- la rotation ϕ_2 suivant l'axe Y_2 ;
- la rotation θ_2 suivant l'axe Z_2.

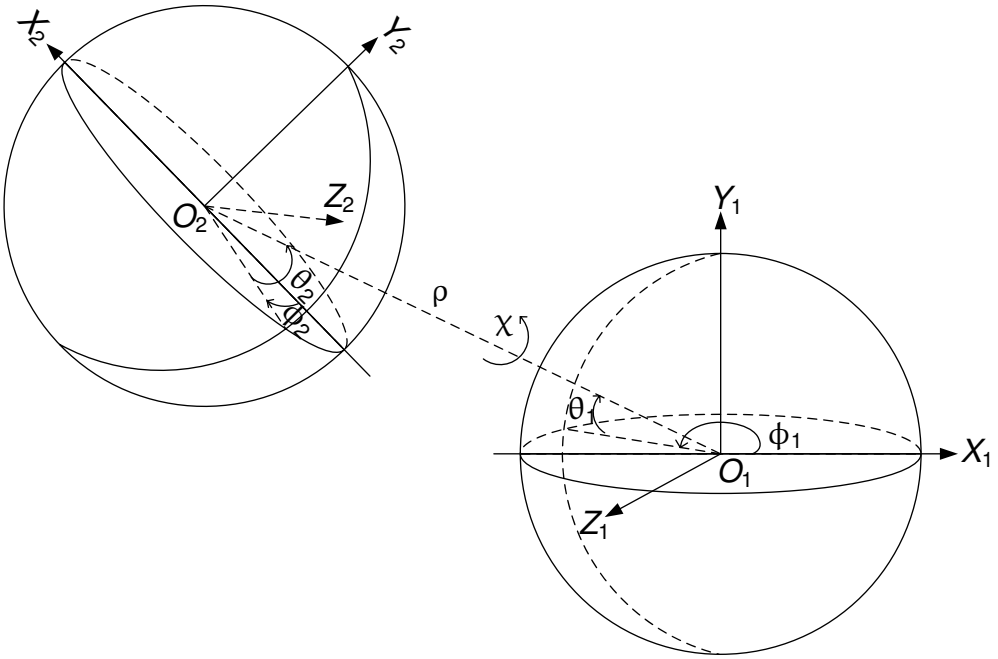


FIGURE 2.1 – L'espace de recherche des candidats d'une interaction entre deux macromolécules biologiques a 6 degrés de liberté : ρ, χ, ϕ_1, θ_1, ϕ_2 et θ_2. Image J.Bernauer, adaptée de [46].

L'exploration quasi-exhaustive de l'espace de recherche n'est pas envisageable dans des temps de calcul raisonnables. Comme le montre Connolly [54, 55, 56], la quantité de candidats à envisager pour tester toutes les combinaisons entre la surface de la protéine et celle de l'ARN est bien trop importante. Sur les 6 degrés de liberté, le nombre de candidats à envisager dépend essentiellement du pas en rotation et en translation, mais aussi de la surface de Connolly de chacun des deux partenaires. La surface de Connolly dépend de la taille d'un partenaire et de sa forme, puisqu'elle reflète la surface accessible au solvant et donc la surface à explorer pour envisager les différentes interactions possibles avec un partenaire. Les différentes interactions possi-

bles avec un partenaire correspondent au nombre de candidats dont la construction est à envisager pour une exploration quasi-exhaustive de l'espace de recherche. Ce nombre de candidats peut dépasser le million pour des partenaires de taille raisonnable (environ 400 acides aminés ou 40 acides nucléiques selon les moyennes constatées sur la PRIDB). La génération des candidats ne s'intéresse donc qu'à un sous-ensemble de candidats. Pour un nombre fixé de candidats à générer, l'objectif est de maximiser les chances d'obtenir des candidats presque-natifs parmi les candidats générés.

En utilisant une méthode de Monte-Carlo, il est possible d'orienter la génération des candidats : plus un candidat a une énergie faible et plus il aura de chances d'être conservé. L'exploration de l'espace des candidats donne alors plus probablement des candidats dans des sous-espaces contenant des candidats de faible énergie. En réitérant un grand nombre de fois la méthode de Monte-Carlo, la génération des candidats permet de maximiser les chances d'obtenir des candidats de faible énergie parmi les candidats générés. En pratique, la réitération de la méthode de Monte-Carlo permet de générer des candidats correspondant à différents minima locaux, potentiellement séparés par des barrières d'énergie importantes. Ces barrières d'énergie, où les candidats ont une énergie très forte, empêcheraient d'appliquer une minimisation d'énergie pour arriver à un candidat presque-natif à partir d'un puits d'énergie différent.

Dans un modèle où les partenaires sont flexibles, il faut, après déplacement d'un partenaire par rapport à l'autre, prendre en compte la déformation des partenaires. Cette déformation s'effectue lorsque l'environnement proche des atomes des partenaires est modifié. La modification de l'environnement proche correspond à la proximité avec des atomes de l'autre partenaire. La proximité entre atomes des deux partenaires définit une interface. Une fois une position déterminée pour le partenaire mobile, la structure 3D des chaînes latérales est donc optimisée en fonction de son environnement proche pour minimiser l'énergie.

Pour minimiser les temps de calcul dans la modification des coordonnées de chaque atome, on utilise les coordonnées internes (voir fig. 2.2) pour chaque acide aminé et chaque acide nucléique. Un arbre des coordonnées des atomes est construit pour chaque acide aminé et chaque acide nucléique :

- chaque atome a 3 atomes parents dans l'arbre ;
- la distance r est la distance au premier parent ;
- l'angle θ donne l'angle formé entre les deux premiers parents et le premier parent avec l'atome, dans le plan formé par les droites des deux premiers et des deux derniers parents ;
- l'angle ϕ est l'angle entre les deux derniers parents et le premier parent avec l'atome, dans le plan perpendiculaire à la droite formée par les deux premiers parents.

Ainsi, modifier les coordonnées de tout un acide aminé ou acide nucléique ne nécessite pas de modifier les coordonnées de l'ensemble de ses atomes. Par ailleurs, les angles θ et ϕ définis chacun pour 2 des rotations d'un partenaire par rapport à l'autre correspondent aux rotations θ et ϕ dans des coordonnées torsionnelles.

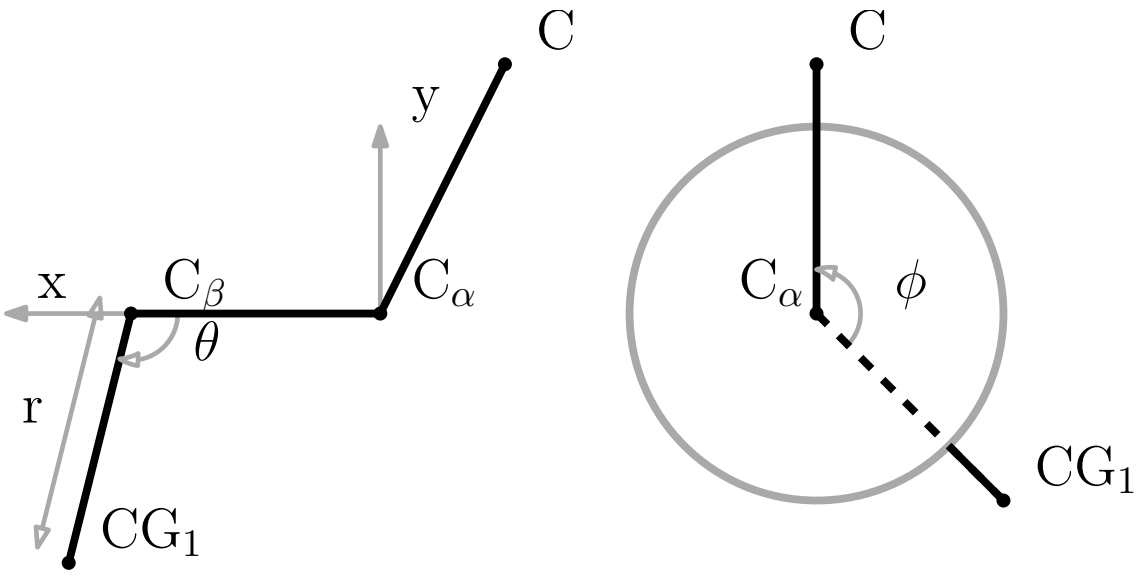


FIGURE 2.2 – La manipulation des coordonnées des structures 3D se fait à l'aide des coordonnées internes : chaque acide aminé et chaque acide nucléique sont représentés sous la forme d'un arbre passant par les liaisons covalentes entre les atomes (les cycles sont coupés). Chaque atome a 3 atomes parents, à l'exception des 3 premiers atomes qui sont définis les uns par rapport aux autres. La position d'un atome est définie par la distance *r* à son premier atome parent et par les angles θ et ϕ. Le référentiel de coordonnées cartésien est donné à titre indicatif. Les atomes donnés en exemple sont *C*, C_α, C_β et l'atome gros-grain CG1.

2.1.1.2 Tri des candidats

L'étape de génération des candidats n'explore pas de manière exhaustive l'espace des conformations possibles entre les deux partenaires étudiés. Toutefois, il est important de noter que, en règle générale, le nombre de candidats devant être générés pour espérer obtenir des candidats presque-natifs est élevé, de l'ordre de 10 000. Certaines fonctions d'évaluation ne sont pas différentiables, ce qui ne permet pas de mettre en place des approches efficaces de minimisation. C'est notamment le cas de la fonction de score gros-grain de RosettaDock, qui somme un terme de score défini par une fonction discrète avec d'autres termes de score (voir section 2.1.2.2).

Sur ce grand nombre de candidats, seul un petit nombre d'entre eux ont une chance d'être des presque-natifs. La proportion est généralement considérée de l'ordre de 1 pour 1 000 pour la grande majorité des complexes, mais il est attendu d'un bon algorithme d'amarrage de dépasser cet ordre de grandeur [114].

Il existe deux types d'erreurs pouvant apparaître suite au tri des candidats. Ils ont des coûts très différents :

– ne pas réussir à mettre un presque-natif en avant face aux leurres oblige à générer plus de candidats pour obtenir un presque-natif dans les premiers candidats du tri ;

– mettre un leurre en avant par rapport aux presque-natifs donne l'illusion d'avoir trouvé un presque-natif, sur lequel peuvent ensuite se baser des expériences qui coûtent cher (par exemple, de cristallographie).

Il est donc impératif de mettre en place d'autres critères d'évaluation les plus efficaces possibles pour éviter ces deux types d'erreurs et identifier les presque-natifs parmi l'ensemble des candidats.

Nous appelons *fonctions de score* les différents critères d'évaluation que nous allons étudier. Selon les besoins, plusieurs fonctions de score peuvent être définies et appliquées. Une première fonction de score agit comme un filtre permettant de supprimer les leurres les plus évidents. Ce filtre est évalué sur l'ensemble des configurations testées pour la génération d'un candidat. Le filtre doit donc être rapide à calculer. Le filtre prend en considération les critères suivants :

– chevauchements des structures ;
– nombre d'atomes impliqués dans l'interaction.

Pour chaque candidat, si les chevauchements sont trop importants ou si le nombre d'atomes impliqués dans l'interaction est trop faible, alors la probabilité qu'il s'agisse d'un leurre est élevée. Le filtre associe alors un score élevé au candidat considéré. Si le candidat considéré a un score trop élevé, il ne fera pas partie des candidats générés. Une seconde fonction de score est définie pour évaluer et optimiser la structure 3D des chaînes latérales à l'interaction. Une telle fonction de score a pour but de donner une structure 3D plus fidèle à la réalité biologique. Grâce à la minimisation des chaînes latérales, les atomes des chaînes latérales voient leurs coordonnées adaptées à l'interaction avec les atomes du partenaire. Un score final est enfin attribué à chaque candidat, pour évaluer sa structure 3D dans son ensemble et trier les candidats en fonction de leur probabilité de représenter correctement l'interaction. Pour calculer ce score, l'ensemble des termes de score est agrégé pour chaque candidat. La fonction d'agrégation est une somme pondérée.

Selon l'objectif du score, ce dernier utilise préférentiellement des paramètres issus de différents modèles géométriques. Un score permettant d'affiner une structure 3D doit pouvoir correctement évaluer l'énergie issue d'interactions avec le solvant, d'où le calcul de l'influence de la solvatation à l'échelle atomique. À l'inverse, un score utilisé comme filtre a besoin d'être calculé rapidement et se focalise préférentiellement sur des paramètres calculés à l'échelle gros-grain.

2.1.1.3 Amarrages gros-grain ou atomique

L'échelle à laquelle s'effectue l'amarrage convient à des problématiques spécifiques. Pour chaque problématique standard d'amarrage, il existe un protocole d'amarrage. L'*amarrage gros-grain* a pour but de découvrir l'*épitope* (ou zone d'interaction), c'est-à-dire identifier les acides nucléiques et acides aminés impliqués dans l'interaction. L'*amarrage atomique* a pour but de déterminer la structure 3D de l'interaction à l'échelle de l'Angström voire en-dessous. Il peut arriver qu'un amarrage atomique soit utilisé lorsque l'on connaît déjà l'épitope, pour affiner la structure 3D identifiée.

Pour réduire les temps de calcul, un amarrage atomique dont on ne connaît pas l'épitope est généralement précédé d'un amarrage gros-grain. Dans un contexte de

prédiction à l'aveugle comme CAPRI, où on ne connait pas la solution, on procède en général à un amarrage atomique à partir des structures candidates générées par l'amarrage gros-grain.

2.1.2 Autres stratégies

Selon le contexte, d'autres stratégies pour affiner l'amarrage local peuvent être envisagées :

- minimisation d'une structure ;
- génération par perturbation d'un nuage de candidats autour d'une structure 3D ;
- évaluation du score d'une structure sans modification de la structure 3D.

La minimisation correspond le plus souvent à une descente de gradient pour obtenir un minimum local proche de la structure 3D donnée en entrée. Une stratégie de minimisation du score permet de comparer les candidats générés par différents outils de génération de candidats, chaque outil ayant une fonction de score différente avec des minima qui ne correspondent pas nécessairement d'un outil à l'autre. Tous les outils de génération de candidats n'ont pas non plus le même seuil à partir duquel deux atomes sont considérés en interaction. La minimisation permet, pour une IRMSD très petite (normalement de moins de 1 Å), de potentiellement diminuer drastiquement l'énergie. Ceci est dû au fait que des contraintes à faible distance peuvent donner de très fortes pénalités aux structures 3D.

La génération par perturbation de candidats autour d'une structure 3D est utilisée, lorsque l'on connaît déjà l'interaction, pour générer des candidats presque-natifs et leurres. Cette stratégie correspond à un amarrage atomique, mais en générant les candidats selon une opération de rotation-translation aléatoire du partenaire mobile, sans utilisation de l'algorithme de Monte-Carlo. De plus, cette stratégie part de la structure native directement, i.e. de la solution, plutôt que de partir de la structure 3D de chacun des deux partenaires. Cette stratégie est davantage détaillée dans la section traitant de la génération de candidats par perturbation (voir section 1.3.2). La génération par perturbation de candidats a déjà été utilisée pour mettre en place et évaluer la fonction de score protéine-protéine de RosettaDock [98].

Évaluer le score d'une structure 3D sans génération de candidats est généralement d'intérêt lorsque l'on compare différentes fonctions de score ou pour évaluer l'énergie d'une structure native. La comparaison des fonctions de score peut avoir lieu à des fins d'évaluation de la structure ou des fonctions de score, comme dans ce manuscrit. L'évaluation de l'énergie d'une structure native permet de vérifier la cohérence d'une fonction de score avec les structures natives disponibles. En effet, une fonction de score correctement modélisée doit idéalement attribuer une énergie plus faible à la structure native qu'aux structures des candidats.

2.1.2.1 Fonctions de score atomique : termes physico-chimiques

Les termes physico-chimiques sont calculés en utilisant des connaissances issues des propriétés physico-chimiques des interactions entre atomes. À l'échelle atomique,

RosettaDock utilise 10 types de termes de score répartis en 6 groupes (voir fig. 2.3). Sur l'ensemble des équations, les mêmes notations sont utilisées pour désigner les mêmes entités, où :

- E est le terme de score, modélisant une énergie physico-chimique ;
- w est le poids donné au terme de score ;
- N est le nombre d'atomes ;
- N_{aa} est le nombre d'atomes gros-grain des acides aminés ;
- N_{na} est le nombre d'atomes gros-grain des acides nucléiques ;
- M est le nombre d'acides aminés ;
- L est le nombre d'acides nucléiques ;
- d est la distance entre deux atomes ;
- P est une probabilité estimée ;
- σ est la distance minimale d'approche entre deux atomes, somme de leurs rayons de Van der Waals [21] ;
- q est une charge positive ou négative ;
- ΔG^{free} est l'énergie de Gibbs [162] ;
- V est le volume atomique [21] ;
- λ est la longueur de corrélation d'un atome [144] ;
- ϕ est l'angle dièdre d'un rotamère formé par le quadruplet d'atomes successifs $CO_0 - NH_1 - C\alpha_1 - CO_1$ [209] ;
- ψ est l'angle dièdre d'un rotamère formé par le quadruplet d'atomes successifs $NH_1 - C\alpha_1 - CO_1 - NH_2$ [209] ;
- ξ, est l'environnement d'un atome.

Les deux termes de score attractif *(fa_atr)* et répulsif *(fa_rep)* de Van der Waals représentent respectivement l'attraction et la répulsion universelle entre atomes. Indépendamment du type d'atome, deux atomes se repoussent s'ils sont trop proches l'un de l'autre. Alternativement, deux atomes s'attirent lorsqu'ils sont suffisamment loin l'un de l'autre, jusqu'à une attraction asymptotiquement nulle au fur et à mesure que la distance croît. Les équations de Lennard-Jones 12-6 présentées ici (voir éq. 2.1 et 2.2) sont une simplification de l'énergie de Van der Waals, où ε_{ij} est la profondeur du puits d'énergie pour i et j. L'énergie de Van der Waals est uniquement calculée pour les atomes à moins de 8 Å de distance. Au-delà de 8 Å, l'énergie de Van der Waals est donc considérée par le modèle comme nulle. Cette approximation permet de minimiser les temps de calcul alloués au calcul de ces deux termes de score. La partie attractive est utilisée pour $0.89\sigma_{ij}$ Å$< d_{ij} < 8$ Å alors que la partie répulsive est utilisée pour $0.6\sigma_{ij}$ Å$< d_{ij} \leq 0.89\sigma_{ij}$ Å.

$$E_{fa\_atr} = w_{fa\_atr} \sum_{i=1}^{N} \sum_{j>i}^{N} \varepsilon_{ij} \left(\left(\frac{\sigma_{ij}}{d_{ij}} \right)^{12} - 2 \left(\frac{\sigma_{ij}}{d_{ij}} \right)^{6} \right) \tag{2.1}$$

$$E_{fa\_rep} = w_{fa\_rep} \sum_{i=1}^{N} \sum_{j>i}^{N} \varepsilon_{ij} \left(\begin{array}{l} \left(\frac{\sigma_{ij}}{0.6\sigma_{ij}} \right)^{12} - 2 \left(\frac{\sigma_{ij}}{0.6\sigma_{ij}} \right)^{6} \\ - \left(0.6\sigma_{ij} - d_{ij} \right) \left(\frac{-12\sigma_{ij}^{12}}{\left(0.6\sigma_{ij} \right)^{13}} + \frac{12\sigma_{ij}^{6}}{\left(0.6\sigma_{ij} \right)^{7}} \right) \end{array} \right) \tag{2.2}$$

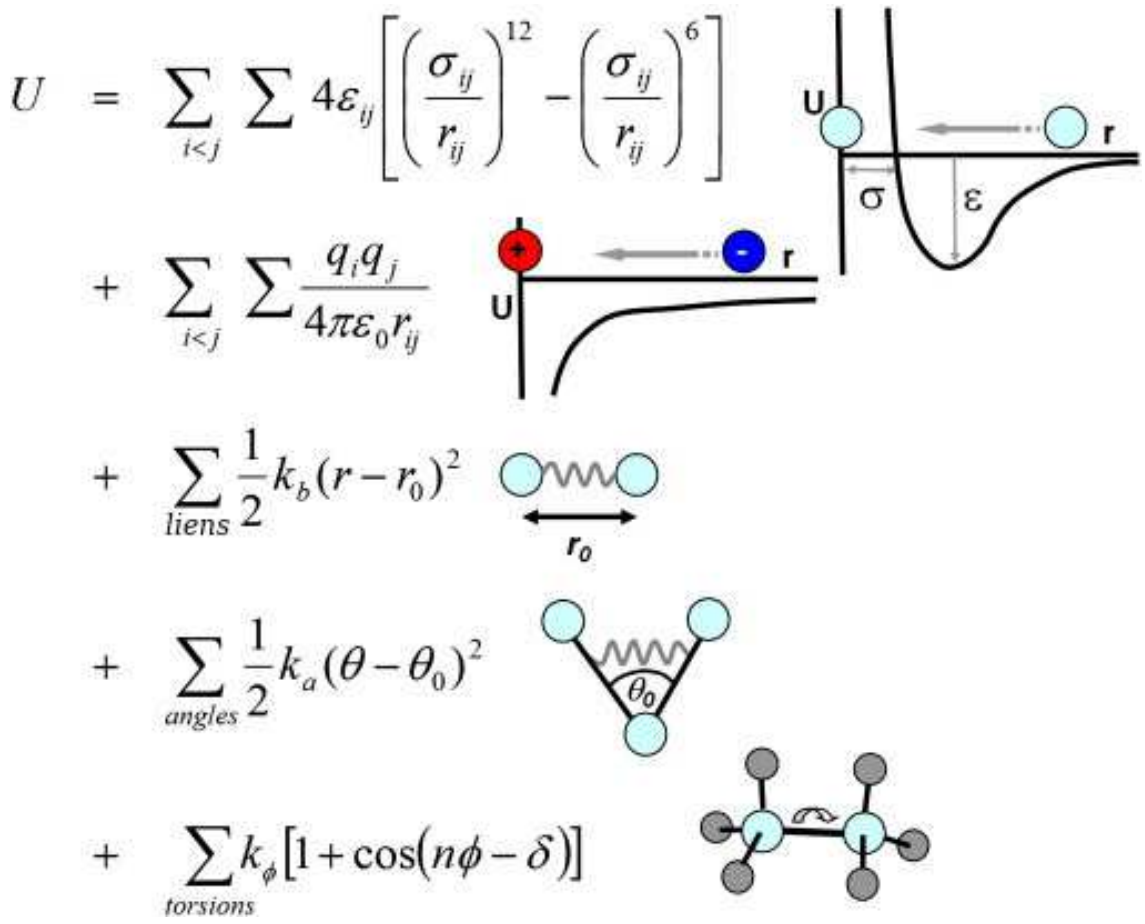


FIGURE 2.3 – L'énergie potentielle d'une structure 3D de macromolécules biologiques, souvent assimilée à un score, est généralement calculée selon une somme de termes modélisant des phénomènes physico-chimiques spécifiques. Les termes peuvent porter sur les interactions covalentes (liens, angles, torsions) ou non covalentes (attraction et répulsion universelles, forces électrostatiques, liaisons hydrogène, *etc.*). Image adaptée de Scientific American / Adv Drug Del Rev.

Dans la cellule, le complexe protéine-ARN est plongé dans un solvant. Il est donc nécessaire que le modèle *in silico* prenne cette caractéristique en compte. Le terme de solvatation *(fa_sol)* est modélisé dans cet objectif. Le terme de solvatation favorise l'accessibilité des parties hydrophiles solvant et l'inaccessibilité des parties hydrophobes au solvant (voir éq. 2.3). Ce terme de score demande le calcul de l'accessibilité au solvant, qui coûte cher en temps de calcul [144].

$$E_{fa\_sol} = w_{fa\_sol} \sum_{i=1}^{N} \sum_{j>i}^{N} \left(\frac{2\Delta G_i^{free} \exp\left(-d_{ij}^2\right)}{4\pi\sqrt{\pi\lambda_i}\sigma_{ij}^2} V_j + \frac{2\Delta G_j^{free} \exp\left(-d_{ij}^2\right)}{4\pi\sqrt{\pi\lambda_j}\sigma_{ij}^2} V_i \right) \tag{2.3}$$

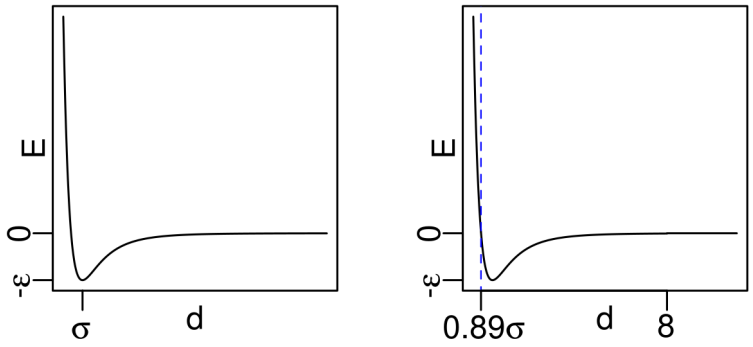


FIGURE 2.4 – Présentation de l'énergie de Van der Waals, où ε correspond à la profondeur du puits d'énergie. L'énergie de Van der Waals possède une part répulsive de 0 Å jusqu'à $0.89\sigma_{ij}$ Å et une part attractive au-delà.

Même sans aller jusqu'à être chargé positivement ou négativement, un atome peut avoir une charge partielle. Cette charge partielle attire préférentiellement l'atome en question vers les atomes ayant une charge partielle opposée. De plus, certains atomes sont plus fréquemment observés à proximité l'un de l'autre qu'avec d'autres atomes. Le terme d'affinité entre paires d'atomes *(fa_pair)* donne un score plus favorable aux atomes ayant une forte affinité entre eux (voir éq. 2.4). Seuls les atomes interagissant entre eux ont une valeur calculée pour fa_pair (*int* dans l'équation). Le terme d'affinité entre paires d'atomes classe chaque atome en quatre environnements, selon sa proximité avec un atome de l'autre partenaire et son accessibilité au solvant. Pour chaque environnement, chaque paire de type d'atomes obtient un score différent. Les quatre environnements possibles sont :

- l'atome est dans l'interaction et accessible au solvant ;
- l'atome est dans l'interaction et inaccessible au solvant ;
- l'atome est hors de l'interaction et accessible au solvant ;
- l'atome est hors de l'interaction et inaccessible au solvant.

$$E_{fa\_pair} = w_{fa\_pair} \prod_{m=1}^{M} \prod_{l=1}^{L} \frac{P(m, l|int, \xi_m, \xi_l)}{P(m|int, \xi_m)\, P(l|int, \xi_l)} \tag{2.4}$$

Outre les affinités qu'il peut y avoir entre des atomes partiellement chargés, les charges, lorsqu'elles sont présentes, ont aussi un impact sur les interactions entre atomes. Le modèle de Coulomb décrit pour cela l'interaction entre atomes chargés en fonction de la valence de leur charge et de la distance qui les sépare. Le terme électrostatique *(hack_elec)* utilise le modèle de Coulomb (voir éq. 2.5).

$$E_{hack\_elec} = w_{hack\_elec} \sum_{i=1}^{N} \sum_{j>1}^{N} \frac{332 q_i q_j}{min\left(d_{ij}, 3\right)^2} \tag{2.5}$$

Un phénomène plus spécifique de charge partielle intervient dans la création et le maintien d'interactions entre molécules et à l'intérieur d'une même molécule. Il s'agit de la formation de liaisons hydrogène. Les liaisons hydrogène se forment entre un hydrogène de charge partielle positive et un autre atome de charge partielle négative (généralement un oxygène ou un azote). Ces liaisons sont des liaisons non covalentes, mais qui participent aux interactions de par leur multiplicité. Les quatre termes de liaisons hydrogène *(hbond_sc, hbond_bb_sc, hbond_lr_bb, hbond_sr_bb)* favorisent l'existence de liaisons hydrogène (voir éq. 2.6). Le score donné aux liaisons hydrogène dépend de la distance d'interaction, à courte portée ou à longue portée, et de la localisation des atomes d'hydrogène, dans le squelette ou dans la chaîne latérale.

$$E_{hbond\_*} = w_{hbond\_*} \sum_{i=1}^{N} \sum_{j>i}^{N} \left(5 \left(\frac{\sigma_{ij}}{d_{ij}} \right)^{12} - 6 \left(\frac{\sigma_{ij}}{d_{ij}} \right)^{10} \right) F(q) \tag{2.6}$$

Les chaînes latérales des acides aminés sont flexibles et peuvent adopter plusieurs conformations. Ces conformations sont catégorisées par probabilité d'apparition dans le diagramme de Ramachandran. La base de données de rotamères de Dunbrack [70] a été conçue sur le même principe pour déterminer, selon la conformation de la chaîne latérale, la probabilité d'apparition correspondante. Le terme de rotamères de Dunbrack *(fa_dun)* donne un score plus favorable aux rotamères les plus présents dans les complexes protéine-protéine connus (voir éq. 2.7). Le terme de rotamères de Dunbrack n'est, pour le moment, utilisé que pour les acides aminés.

$$E_{fa\_dun} = w_{fa\_dun} \sum_{m=1}^{M} - \ln \left(P_{type(m)} (\phi_m \psi_m) \right) \tag{2.7}$$

2.1.2.2 Termes physico-chimiques à l'échelle gros-grain

À l'échelle gros-grain, RosettaDock calcule quatre types de termes de score. Les atomes considérés sont alors des atomes gros-grain. Dans cette section, on note N le nombre d'atomes gros-grain.

Le terme de contact *(interchain_contact)* est une version simplifiée du terme d'attraction de Van der Waals. Le score donné au terme de contact est une fonction discrète dépendant du nombre n_{Int} d'atomes gros-grain à l'interface. Cette fonction discrète est définie par une droite avec une valeur extrême et deux points particuliers. S'il n'y a aucun atome gros-grain en interaction, le terme de contact vaut 12. S'il y a 2 atomes gros-grain en interaction, il vaut 9.5. Entre 3 et 19 atomes gros-grain, il vaut $10 - 0.5 * n_{Int}$. Au-delà de 19 atomes gros-grain, il est égal à sa valeur extrême : -10.

Le terme de répulsion de Van der Waals gros-grain *(interchain_vdw)* est aussi une version simplifiée du terme de répulsion de Van der Waals atomique. Le terme de répulsion de Van der Waals gros-grain vaut entre 0 et 1. Il utilise un paramètre $d_{\alpha,\beta}$ déterminé au moyen d'une mesure statistique sur un jeu de données de complexes, où α est le type de l'atome i et β le type de l'atome j (voir éq. 2.8). Nous avons notamment

déterminé les valeurs de ce paramètre pour chaque paire (α, β) de type d'atomes gros-grain.

$$E_{interchain\_vdw} = w_{interchain\_vdw} \sum_{i=1}^{N_{aa}} \sum_{j=1}^{N_{na}} \frac{\left(d^2_{\alpha,\beta} - d^2_{ij}\right)}{d^2_{\alpha,\beta}} \tag{2.8}$$

Le terme d'affinité par paire d'atomes gros-grain *(interchain_pair)* fonctionne de la même manière que le terme d'affinité par paire d'atomes.

Le terme d'environnement *(interchain_env)* reprend le principe d'environnement des termes d'affinité par paire d'atomes. Le terme d'environnement donne un score favorable aux atomes gros-grain se trouvant dans les environnements où ils ont le plus de chances d'être trouvés. Le terme d'environnement prend aussi en compte une version simplifiée du terme de solvatation.

2.2 Évaluation de la fonction de score non optimisée

La fonction de score non optimisée pour les complexes protéine-ARN est la fonction de score par défaut dans RosettaDock. Cette fonction de score est directement issue de l'amarrage protéine-protéine : elle est optimisée pour l'amarrage protéine-protéine. Nous appellerons cette fonction de score *ROS*. Nous allons d'abord observer les mesures d'évaluation globales avant de traiter les mesures d'évaluation locales.

2.2.1 Mesures d'évaluation globales

Les mesures d'évaluation globales peuvent s'appliquer lorsque l'on dispose de la matrice de confusion. Calculer la matrice de confusion nécessite de déterminer un seuil à partir duquel la fonction de score sépare en presque-natifs et en leurres les candidats prédits. Nous utilisons deux seuils différents. Le premier seuil est le seuil permettant d'obtenir, pour chaque complexe, le meilleur Fscore (voir section 0.6.2). Ce seuil permet de savoir à quel point la fonction de score peut trouver un compromis entre deux types d'erreurs : prédire presque-natif un leurre (Faux Positif) et ne pas prédire presque-natif un presque-natif (Faux Négatif).

Avec ce seuil, le nombre de candidats prédits presque-natifs est très élevé (voir tableau 2.1). Le rappel est très élevé, mais on peut remarquer avec la précision que la moitié des candidats prédits sont des presque-natifs. En moyenne, chaque candidat prédit presque-natif a donc une chance sur deux d'être un presque-natif. Et le rappel montre que la quasi-totalité des presque-natifs sont prédits presque-natifs. De plus, la spécificité continue de montrer que la fonction de score ROS n'est pas adaptée à la prédiction d'interactions protéine-ARN. Pour plus de la moitié des complexes, les leurres prédits par la fonction de score ROS représentent moins de 1 % des véritables leurres. Dit autrement, plus de 99 % des leurres sont prédits comme étant des presque-natifs.

Aggrégat	Précision	Rappel	Fscore	*Accuracy*	Spécificité	Seuil
Moyenne	49.52%	98.68%	65.94%	50.75%	2.27%	9 874
Médiane précision	47.44%	99.99%	64.35%	47.45%	0.06%	9 998
Médiane Fscore	46.51%	99.99%	63.49%	46.53%	0.06%	9 997

TABLE 2.1 – Évaluation globale de la fonction de score ROS sur la PRIDB, sous le seuil du meilleur Fscore pour chaque complexe, avec la moyenne et la médiane sur les 120 complexes de la PRIDB. La dernière colonne correspond au nombre de candidats prédits presque-natifs au seuil du meilleur Fscore et permet de comparer ces résultats à ceux du top10. Pour le calcul des mesures des lignes concernant la médiane, ce sont les exemples correspondant à la médiane de la précision, d'une part, et du Fscore, d'autre part, qui sont utilisés.

L'évaluation de la fonction de score par des mesures globales avec un seuil défini par le Fscore donne des résultats apparemment satisfaisants. Mais le seuil est alors différent pour chaque complexe et nécessite de connaître à l'avance le résultat. On ne peut donc pas choisir ce seuil, qui est généralement très élevé, avec la plupart des valeurs au-delà de 9 990 candidats prédits presque-natifs. Le seuil défini par le meilleur Fscore est uniquement étudié pour mesurer un compromis de performance de la fonction de score. De plus, même si l'on pouvait trouver ce seuil de compromis entre précision et rappel, la précision n'est que de moitié. Cela signifie que, en prenant au hasard un candidat parmi les candidats prédits, il y a en moyenne 50 % de chances que le candidat prédit ne soit pas un presque-natif. Or, l'objectif est justement de nous assurer de pouvoir trouver au moins un presque-natif, avec comme marge de manœuvre le fait de pouvoir en sélectionner 10.

Le second seuil est fixé à 10 candidats, ce qu'on appelle le top10. Ainsi, seuls les 10 meilleurs candidats au sens du tri opéré par la fonction de score sont considérés comme des presque-natifs. Tous les autres candidats sont prédits comme étant des leurres.

Avec un seuil au top10, on ne s'attend pas à avoir un fort rappel (ou sensibilité) lorsque le nombre de presque-natifs est grand dans l'ensemble des exemples testés. Rappelons que le nombre de presque-natifs est d'au moins 30 pour chaque complexe de la PRIDB. Et le rappel est effectivement très bas (voir tableau 2.2). Mais surtout, la précision est aussi très basse, avec 59 des 120 complexes - soit près de la moitié d'entre eux - n'ayant aucun presque-natif dans le top10 (voir tableau S2).

On voit donc que les mesures d'évaluation globales affichent parfois des résultats très satisfaisants, avec un rappel ou une spécificité très forte. Mais les résultats ne correspondent en réalité pas aux objectifs fixés, ce que peut montrer notamment la précision. C'est pourquoi il est utile de passer à des mesures d'évaluation locales.

Aggrégat	Précision	Rappel	Fscore	*Accuracy*	Spécificité
Moyenne	29.25%	0.05%	0.09%	68.17%	99.83%
Médiane précision	10.00%	0.06%	0.12%	81.13%	99.89%
Médiane Fscore	15.00%	0.02%	0.04%	26.26%	99.28%

TABLE 2.2 – Évaluation globale de la fonction de score ROS sur la PRIDB, sous le seuil de 10 candidats (top10), avec la moyenne et la médiane sur les 120 complexes de la PRIDB. Pour la médiane, ce sont les complexes correspondant à la médiane de la précision, d'une part, et du Fscore, d'autre part, qui sont utilisés pour calculer les différentes mesures, dont les lignes sont intitulées précision et Fscore.

2.2.2 Mesures d'évaluation locales

Les mesures d'évaluation locales sont au moins pour certaines plus spécifiques au problème étudié et plus adaptées aux objectifs fixés pour la prédiction d'interactions protéine-ARN. Les mesures d'évaluation locales se déclinent en deux axes :

- d'une part, des mesures sur les performances de la fonction de score, comme l'aire sous la courbe ROC ;
- d'autre part, des mesures davantage focalisées sur les objectifs biologiques de la prédiction, comme le nombre de presque-natifs dans le top10 des candidats, le score d'enrichissement et le diagramme d'énergie en fonction du IRMSD.

Les aires sous la courbe ROC sont en moyenne de 0.37, avec une variance de 0.02. On pourrait penser qu'inverser la prédiction donnée par la fonction de score ROS donnerait de meilleurs résultats. Inverser la prédiction correspond par exemple à attribuer à chaque candidat le négatif de son score, ce qui inverse l'ordre dans lequel les candidats se situent dans un tri selon le score. Pour l'aire sous la courbe ROC, cela aurait pour conséquence d'obtenir le miroir par rapport à 0.5 de la valeur de l'aire sous la courbe de ROC (ici, 0.63, puisque $0.5 - 0.37 = 0.13$ et $0.5 + 0.13 = 0.63$). Cependant, c'est sans compter sur le fait que la proportion de presque-natifs dans un ensemble de candidats est bien plus faible pour une prédiction à l'aveugle (de l'ordre de 1 à 10 pour 10 000 avec un bon protocole de génération de candidats). Inverser la prédiction reviendrait certainement à donner une énergie élevée à une grande partie des presque-natifs de faible énergie.

De la même manière, avec 59 complexes sur 120, près de la moitié des complexes n'ont aucun presque-natif dans le top10 des meilleurs candidats en énergie. Formulé autrement, en prenant un complexe au hasard de la PRIDB, il y a une chance sur deux que la prédiction ne propose aucun candidat satisfaisant. La fonction de score ROS ne permet pas la prédiction des interactions protéine-ARN.

Les diagrammes d'énergie en fonction du IRMSD permettent de constater que la fonction de score ROS n'est pas adaptée à un raffinement d'une structure de haute résolution protéine-ARN. En effet, aucun des complexes ne montre un entonnoir sur la totalité de ses candidats : il existe toujours de nombreux candidats de faible énergie et ayant un IRMSD élevé (voir 2.5). On peut aussi constater plus clairement qu'inverser la fonction de score ROS amènerait à n'avoir qu'une très faible quantité de candidats de

faible énergie, souvent avec un IRMSD supérieur à 5 Å.

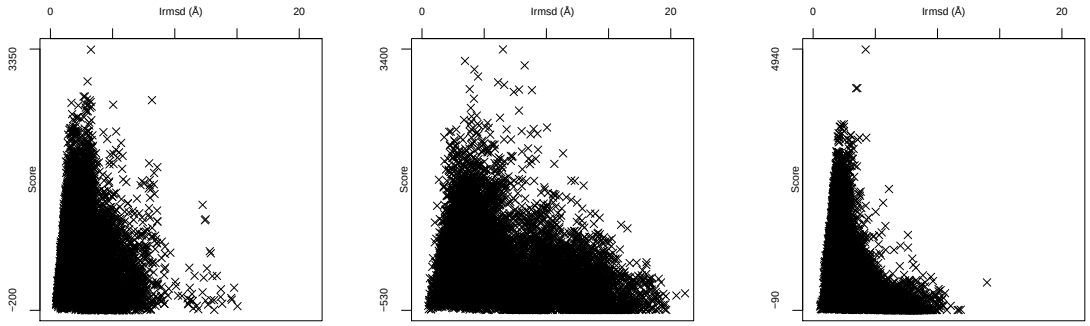


FIGURE 2.5 – Exemples de diagrammes d'énergie en fonction du IRMSD pour la fonction de score ROS sur 3 complexes. On remarque l'absence d'entonnoir. De nombreux complexes montrent des diagrammes semblables pour la fonction de score ROS, qui n'est pas adaptée.

On note toutefois que certains complexes peuvent être considérés comme ayant un entonnoir, jusqu'à une faible valeur de IRMSD (voir 2.6). Au-delà de cette valeur en IRMSD, de l'ordre de 2 ou 3 Å et propre à chacun de ces complexes, la fonction de score ROS donne une énergie comparable entre des candidats ayant un IRMSD faible comme élevé.

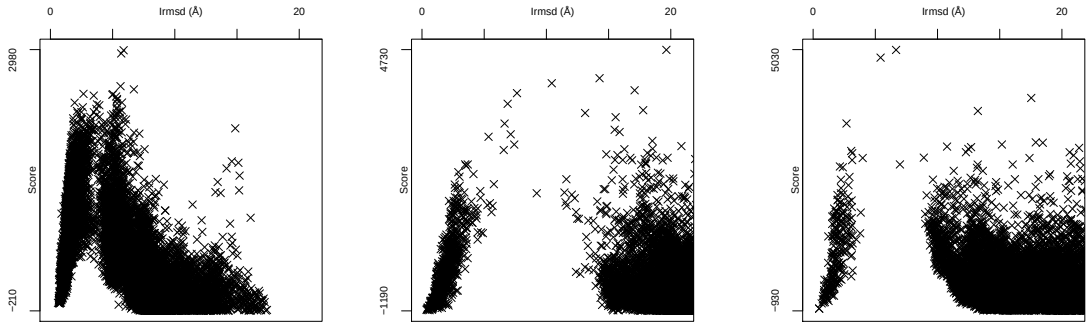


FIGURE 2.6 – Meilleurs exemples de diagrammes d'énergie en fonction du IRMSD pour la fonction de score ROS sur 3 complexes. On remarque un entonnoir s'arrêtant rapidement. Rares sont les complexes montrant des diagrammes semblables pour la fonction de score ROS.

Si la fonction de score non optimisée ROS n'est pas adaptée à la prédiction d'interactions protéine-ARN, il est possible d'optimiser une fonction de score similaire. Nous verrons lors de l'optimisation de la fonction de score, avec le filtrage *a priori*, que ROS est initialement apprise pour prédire des candidats dont certains leurres sont

déjà filtrés. Il s'agit notamment des candidats avec trop d'interpénétrations, aussi appelées *clashes*, notion modélisée par le terme de score fa_rep. Ce constat, couplé au fait que ROS n'a pas été construite pour prédire d'interactions protéine-ARN, explique son échec sur la prédiction d'interactions protéine-ARN. Mais nous verrons aussi et surtout d'autres approches d'optimisation.

2.3 Optimisation de la fonction de score

La combinaison linéaire de termes physico-chimiques est la seule forme de fonction de score que RosettaDock peut nativement utiliser. Nous mettons donc en œuvre l'apprentissage d'une fonction de score étant la combinaison linéaire des 10 attributs physico-chimiques atomiques disponibles pour chaque candidat. L'équation 2.9 montre pour un candidat c la combinaison linéaire $E(c)$ des $|A|$ attributs physico-chimiques a_i. w_{a_i} représentent les poids de la fonction de score et $E_{a_i}(c)$ la valeur de l'attribut a_i pour le candidat c.

$$E(c) = \sum_{i=1}^{|A|} w_{a_i} E_{a_i}(c) \tag{2.9}$$

Une autre formulation envisageable est celle présentée dans l'équation 2.10 avec des notations plus classiquement utilisées en apprentissage. On retrouve X qui désigne le candidat c, x_i qui représente la valeur de l'attribut E_{a_i} pour le candidat c ($E_{a_i}(c)$) et w_i le poids du i^{eme} attribut.

$$E(X) = \sum_{i=1}^{|A|} w_i x_i \tag{2.10}$$

Par souci de simplicité et pour éviter d'introduire un formalisme trop distant des conventions usuelles en apprentissage automatique, nous utiliserons les notations présentées dans l'équation 2.10.

Les poids w_i doivent être définis soit à partir de connaissances physico-chimiques, soit à partir de données déjà connues, c'est-à-dire des complexes protéine-ARN pour lesquels un ensemble de presque-natifs et un ensemble de leurres sont connus. Pour conserver la sémantique biologique de la fonction de score qui représente une énergie, les presque-natifs doivent avoir des scores de valeurs inférieures à ceux des leurres.

Si nous arrivons à construire une telle fonction de score, nous pourrons l'utiliser à des fins prédictives. C'est-à-dire que seules les candidats (ou exemples) ayant les valeurs les plus faibles pour ce score seront retenus.

Du point de vue de l'apprentissage, cela implique qu'il existe une corrélation entre la probabilité d'être un candidat presque-natif et la valeur du score. Nous pouvons donc estimer la probabilité conditionnelle $P(Presque-natif|E(X))$ et de manière plus générale $P(Presque-natif|X)$.

Dans notre cadre de travail, nous disposons de deux classes : presque-natif et leurre. Nous noterons 1 la classe presque-natif et 0 la classe leurre.

Notre objectif se reformule donc comme l'estimation de la probabilité conditionnelle : $P(y = 1|X)$.

Compte tenu du cadre imposé par RosettaDock, nous cherchons à estimer $P(y = 1|X) \sim E(X) = \sum_{i=1}^{|A|} w_i x_i$. Nous devons donc déterminer les poids w_i tels que $E(X)$ représente bien la probabilité pour X d'être un presque-natif. Pour être tout à fait rigoureux, sachant que $E(X)$ représente (d'un point de vue biologique) l'énergie du candidat X, plus cette énergie est faible alors plus la probabilité que le candidat X soit un presque-natif est élevée.

Nous cherchons donc à déterminer les poids w_i tels que $P(y = 1|X) = 1 - E(X) = 1 - \sum_{i=1}^{|A|} w_i x_i$ aient une valeur maximale pour X presque-natif.

Plusieurs approches peuvent être mises en œuvre pour estimer les poids w_i.

Nous présentons dans les sous-sections 2.3.1 et 2.3.2 deux approches permettant de déterminer ces poids.

2.3.1 Estimation des poids par régression logistique

Nous allons montrer dans cette section le bien fondé de la modélisation de $P(y = 1|X)$ par une combinaison linéaire des attributs de X

Sachant que notre objectif est de pouvoir identifier les presques-natifs, nous pouvons redéfinir notre objectif principal : *estimer au mieux le ratio* $P(y = 1|X)/P(y = 0|X)$. Si le ratio est supérieur ou égal à 1 alors le candidat sera prédit comme étant un presque-natif, sinon, il sera prédit comme étant un leurre.

Par un simple jeu de réécriture des probabilités conditionnelles, nous pouvons reformuler le rapport $P(y = 1|X)/P(y = 0|X)$ comme indiqué dans l'équation 2.11.

$$\frac{P(y = 1|X)}{P(y = 0|X)} = \frac{P(y = 1)}{P(y = 0)} * \frac{P(X|y = 1)}{P(X|y = 0)} \tag{2.11}$$

Classiquement la fraction $\frac{P(y=1)}{P(y=0)}$ est estimée à partir du rapport des contingences observées sur le jeu d'apprentissage : $\frac{P(y=1)}{P(y=0)} = \frac{nb(y=1)}{nb(y=0)}$ où $nb(y = 1)$ (resp. = 0) représente le nombre de candidats presque-natifs (resp. leurres) dans le jeu d'apprentissage.

L'estimation du rapport des probabilités conditionnelles exprimé dans l'équation 2.11 repose donc sur l'estimation du rapport défini dans l'équation 2.12.

$$\frac{P(X|y = 1)}{P(X|y = 0)} \tag{2.12}$$

Pour estimer le rapport 2.12, la régression logistique pose l'hypothèse fondamentale que le logarithme népérien du rapport des probabilités conditionnelles peut s'exprimer comme une combinaison linéaire des attributs décrivant X (voir équation 2.13).

$$\ln\left(\frac{P(X|y = 1)}{P(X|y = 0)}\right) = a_0 + \sum_{i=1}^{|A|} a_i x_i \tag{2.13}$$

Sachant que nous cherchons à estimer la probabilité d'un candidat d'être un presque-natif, nous devons estimer $P(y = 1|X)$.

Nous pouvons obtenir une estimation de $P(y = 1|X)$ grâce à l'équation 2.14 avec $w_0 = ln\frac{P(y=1)}{P(y=0)} + a_0$ et $w_i = a_i$.

$$\text{LOGIT}(P(y = 1|E)) = \ln \frac{P(y = 1|X)}{1 - P(y = 1|X)} = w_0 + \sum_{i=1}^{|A|} w_i E_i \tag{2.14}$$

L'estimation de $P(y = 1|X)$ est fournie par l'équation 2.15.

$$P(y = 1|X) = \frac{e^{w_0 + \sum_{i=1}^{|A|} w_i x_i}}{1 - e^{w_0 + \sum_{i=1}^{|A|} w_i x_i}} \tag{2.15}$$

Rappelons que notre objectif est de déterminer les poids w_i permettant d'estimer au mieux $P(y = 1|X)$. Plusieurs méthodes peuvent être utilisées pour déterminer ces poids. Classiquement en régression logistique, ces poids sont estimés par maximum de vraisemblance à partir d'un échantillon de données considéré comme représentatif.

Nous ne détaillerons pas les calculs nécessaires pour l'estimation par maximum de vraisemblance. Le lecteur intéressé par le maximum de vraisemblance appliqué au raffinement de structures 3D pourra se référer à [184].

D'autres approches peuvent être utilisées pour estimer les valeurs des poids, par exemple des approches d'optimisation stochastiques telles que les algorithmes de type évolutionnaire.

2.3.2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires appartiennent à la famille des méthodes d'optimisation stochastiques. Ces algorithmes permettent d'obtenir une solution approchée à un problème.

Dans notre cadre de travail, notre objectif est de pouvoir déterminer, à partir d'un ensemble d'attributs associés à un candidat, si celui-ci est un presque-natif ou non. Si nous nous replaçons dans le cadre de RosettaDock, nous cherchons à déterminer les poids w_i tels que $E(X) = \sum_{i=1}^{|A|} w_i x_i$ soit minimal si X est un presque-natif.

Pour déterminer les poids w_i, nous avons vu que nous pouvons estimer ces poids par maximum de vraisemblance, mais nous pouvons aussi déterminer ces poids par rapport à une fonction objectif et un jeu de données étiquetées. Et comme indiqué précédemment, notre objectif est de déterminer les poids w_i tels que les presque-natifs obtiennent des valeurs faibles pour E et les leurres des valeurs élevées.

Nous avons choisi de mettre en œuvre l'approche ROGER *(ROC-based Evolutionary learneR)* [228]. Il s'agit d'une implémentation en C d'un algorithme génétique cherchant à optimiser un vecteur de valeurs selon une fonction objectif. ROGER a pour fonction objectif l'aire sous la courbe *Receiver Operating Characteristic* (ROC-AUC), à maximiser. En pratique, ROGER minimise la somme des rangs des exemples positifs, qui est un équivalent de la maximisation de l'aire sous la courbe ROC.

Le principe de ROGER (principe général des algorithmes évolutionnaires) est illustré dans la figure 2.7.

Le principe consiste à manipuler des individus génétiques qui vont évoluer dans l'espace de recherche au moyen de mutations et de croisements de leur patrimoine génétique.

Plusieurs espaces de recherche peuvent être définis. Dans notre cadre de travail, l'espace de recherche est $\mathbb{R}^{|A|}$ car les $|A|$ attributs caractérisant un candidat sont à valeur réelle.

Un individu génétique est défini comme l'ensemble des $|A|$ poids w_i que nous cherchons à définir. Étant donné un individu génétique, tous les candidats du jeu d'apprentissage sont évalués et le score obtenu pour chaque individu permet d'ordonner tous les candidats par valeur croissante de ce score (E tel que défini dans l'équation 2.10).

La qualité d'un individu génétique f est évaluée à partir de ce tri. Cette qualité se définit simplement par la somme des rangs des exemples presque-natifs.

Le principe général de ROGER comporte globalement les cinq étapes suivantes (voir fig. 2.7) :

1. génération aléatoire de la population initiale des individus génétiques (μ parents), puis évaluation de cette population ;
2. vérification de la qualité de la population. Fin de l'algorithme si au moins l'un des critères d'arrêt est atteint ;
3. évolution de cette population pour obtenir un ensemble d'enfants de cardinalité λ ; La génération de ces λ enfants est réalisée par mutation et croisement à partir des μ parents de la population courante ;
4. évaluation de l'ensemble des λ nouveaux enfants ;
5. sélection de la nouvelle population, ne conservant que les μ meilleurs individus génétiques à partir de l'ensemble des individus de la population courante, *i.e.* les μ parents + λ enfants.

Nous avons testé différents couples de valeurs pour μ et λ. Compte tenu des performances obtenues et des temps de calculs, le meilleur couple de valeurs est $(\mu, \lambda) = (10, 80)$.

Nous avons défini trois critères d'arrêt pour l'algorithme :
- le nombre maximum d'itérations est atteint (100 000 itérations) ;
- le nombre maximum d'itérations sans amélioration de la fonction objectif est atteint (20 000 itérations) ;
- l'optimum de la fonction objectif est atteint, c'est-à-dire que l'aire sous la courbe ROC est égale à 1.

ROGER peut apprendre différents types de fonctions parmis lesquels nous retrouvons des fonctions linéaires, polynomiales, logistiques, *etc*.

2.3.3 Méthodologie d'optimisation de la fonction de score atomique

La méthodologie générale d'optimisation de la fonction de score atomique de RosettaDock peut se décomposer en 3 parties :

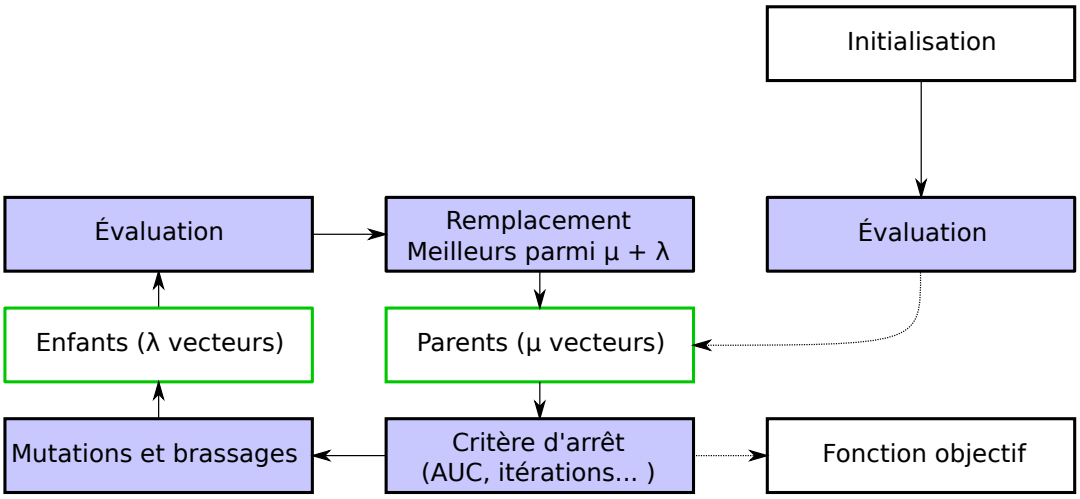


FIGURE 2.7 – Illustration de l'algorithme général de ROGER. Les parents et enfants que ROGER manipule sont des vecteurs de valeurs. En encadrés verts se trouvent les vecteurs. Les étapes du processus sont encadrées en bleu et l'entrée et la sortie sont en encadrés noirs. ROGER commence par choisir au hasard μ parents et les évaluer. Puis, il génère λ enfants par mutation et recombinaison à partir des μ parents. ROGER évalue et sélectionne ensuite selon la fonction objectif les μ meilleurs vecteurs parmi les μ parents et λ enfants. Ces μ meilleurs vecteurs deviennent les μ parents de l'itération suivante, jusqu'à atteindre un critère d'arrêt.

– la constitution des jeux de données, déjà présentée au chapitre 1 ;
– l'apprentissage de la fonction de score sur l'ensemble des 120 complexes et l'apprentissage des 120 fonctions de score en *leave-"one-pdb"-out*, présentés précédemment dans cette même section ;
– l'évaluation des 120 fonctions de score en *leave-"one-pdb"-out* sur les candidats de leur jeu d'évaluation et celle de la fonction de score sur les 40 000 candidats de l'ensemble de validation.

Sur le diagramme 2.8, la première partie est délimitée par les points *a*, *b* et *c*. Ces points *a*, *b* et *c* correspondent respectivement à la génération des candidats par perturbation, à leur étiquetage en presque-natifs, leurres et leurres du test et à l'échantillonnage sans remise des candidats des classes presque-natifs et leurres pour constituer le jeu d'apprentissage. La seconde partie intervient aux points *d* et *e* dans la construction des fonctions de score, qu'il s'agisse des 120 fonctions de score apprises en *leave-"one-pdb"-out* – point *d* – ou de la fonction de score apprise sur l'ensemble des 120 complexes – point *e*. La dernière partie est représentée par le point *f*, avec le calcul et l'observation des différentes mesures d'évaluation sur les exemples de validation.

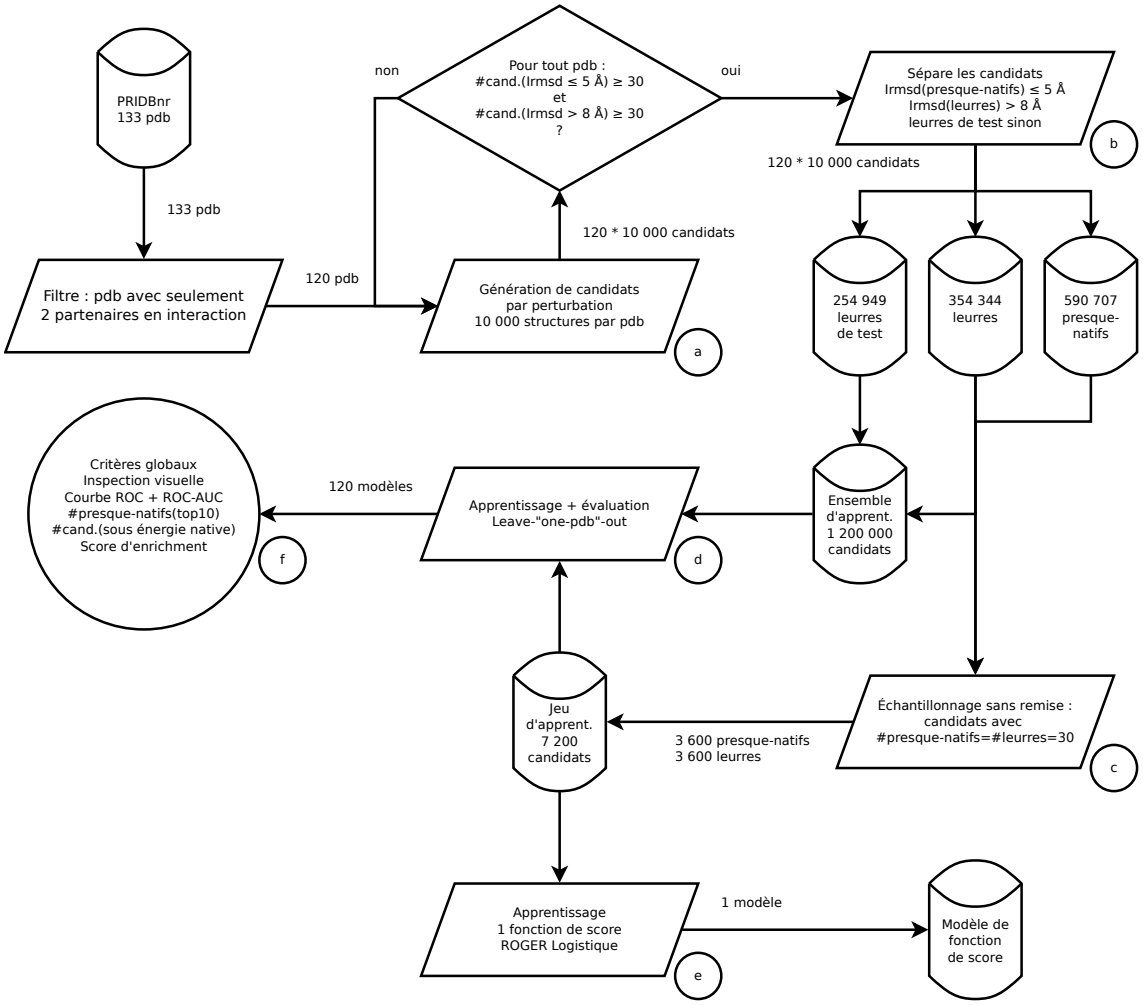


FIGURE 2.8 – Illustration de la méthodologie d'optimisation de la fonction de score atomique.

2.4 Évaluation de la fonction de score optimisée

Le modèle utilisé pour l'apprentissage et présenté précédemment peut faire l'objet de variantes pour améliorer le pouvoir prédictif tel qu'il est reflété par les différentes mesures d'évaluation.

Les choix sur les variantes dépendent des connaissances *a priori* sur le contexte de la prédiction protéine-ARN. D'une part, nous choisissons de tester les méthodologies mises en œuvre dans la prédiction d'interactions protéine-protéine. D'autre part, nous prenons en compte les caractéristiques propres des ARN par rapport aux protéines, notamment sa plus grande flexibilité. Parmi ces choix, nous pouvons identifier :

– la définition de l'intervalle de valeurs des poids des termes de score ;
– le filtrage *a priori* des candidats pour chaque complexe ;
– la catégorisation des complexes en fonction de critères sur leurs partenaires

combinée à la modélisation d'une fonction de score par catégorie de complexes.

2.4.1 Pouvoir prédictif en fonction des contraintes

L'intervalle de valeurs dans lequel les poids des termes de score sont recherchés influence les performances de la fonction de score obtenue pour maximiser la fonction objectif. Une contrainte imposée sur cet intervalle de valeurs permet de simplifier la recherche du maximum global en diminuant la taille de l'espace de recherche. Mais contraindre l'intervalle de valeurs des poids peut aussi empêcher de trouver les poids permettant d'obtenir le maximum global. Il convient donc d'utiliser une contrainte astucieusement choisie pour éviter les minima locaux tout en conservant dans l'intervalle de valeurs les poids permettant d'obtenir le maximum global.

Quelles que soient les contraintes appliquées à l'apprentissage dans ROGER, les fonctions de score optimisées pour l'amarrage protéine-ARN surpassent les performances de la fonction de score optimisée pour l'amarrage protéine-protéine qui est, rappelons-le, la fonction par défaut de RosettaDock (voir tableau 2.3). La fonction de score optimisée pour l'amarrage protéine-protéine a une AUC moyenne inférieure à 0.4 et n'a que la moitié des complexes avec un nombre de presque-natifs non nul parmi le top10 des candidats triés en énergie. De plus, seuls 15 des complexes ont une AUC supérieure à 0.5.

La contrainte à poids positifs, issue de la connaissance experte de l'amarrage protéine-protéine, donne de meilleurs résultats que la contrainte à poids négatifs ou même que l'absence de contrainte [101]. Les fonctions de score à poids positifs ont une ROC-AUC moyenne supérieure de plus de 0.1 à la ROC-AUC moyenne des autres fonctions de score. De plus, les fonctions de score à poids positifs ont une dizaine de complexes de plus avec une ROC-AUC supérieure à 0.5 et avec un nombre de presque-natifs non nul dans le top10 des candidats triés en énergie.

La contrainte à poids négatifs donne des résultats quasi-identiques à l'absence de contrainte. Ces deux manières d'apprendre les fonctions de score donnent une ROC-AUC moyenne de 0.64, 105 complexes avec une ROC-AUC supérieure à 0.5 et 109 complexes avec au moins 1 presque-natif dans le top10 des candidats triés en énergie. En l'absence de contrainte, les poids sont piégés dans des minima locaux. Ce sont justement ces minima locaux qui sont évités en appliquant la contrainte des poids positifs. Étant donnée la similarité de résultats entre la contrainte négative et l'absence de contrainte, la question est de savoir si les poids sont aussi similaires.

Il se trouve que les minima locaux des fonctions de score sans contrainte appliquée sur leurs poids sont pour certains négatifs ou nuls (voir fig. 2.9). Les poids en question des fonctions de score sans contrainte sont donc effectivement piégés dans l'intervalle de valeurs négatif.

Les poids finalement acceptés ont surtout des valeurs non nulles pour quatre termes de score (voir fig. 2.10). Parmi ces quatre termes de score, trois d'entre eux correspondent à des types de liaisons hydrogène et ont une valeur proche de 1 pour quasiment tous les complexes. Ceci montre à quel point la formation des liaisons hydrogène dans une interaction protéine-ARN est importante dans les structures biologiquement

Classifieur	ROC-AUC	ROC-AUC > 0.5	#(top10$_E$(Candidats))> 0
ROS	0.363$\pm$0.016	15	61
NEG	0.640$\pm$0.016	105	109
ALL	0.643$\pm$0.016	105	109
POS	**0.798**$\pm$0.018	**119**	**117**

TABLE 2.3 – Pour chaque intervalle de valeurs utilisé comme contrainte d'apprentissage des poids par ROGER linéaire sont indiqués : la moyenne et la variance de la ROC-AUC, le nombre de complexes pour lesquels la ROC-AUC est supérieure à 0.5 et le nombre de complexes avec un #presque-natifs (top10$_E$(Candidats)) non nul. ROGER linéaire positif (POS) a pour contrainte l'intervalle de valeurs positif [0 ; 1], ROGER linéaire (ALL) l'intervalle de valeurs sans contrainte [-1 ; 1] et ROGER linéaire négatif (NEG) l'intervalle de valeurs négatif [-1 ; 0]. La fonction de score par défaut est celle implémentée dans RosettaDock et optimisée pour l'amarrage protéine-protéine.

actives. Le seul type de liaison hydrogène qui n'est pas privilégié correspond aux liaisons hydrogène entre les atomes de la chaîne latérale de l'un des partenaires et les atomes du squelette de l'autre partenaire. Le dernier terme de score a une valeur qui avoisine les 0.2 et correspond à l'affinité entre paires d'atomes.

2.4.2 Fonction de score dédiée pour chaque type d'ARN

Une des hypothèses de travail est de considérer que l'interaction entre la protéine et l'ARN dépend de la forme des partenaires. Dans cette optique, nous avons catégorisé les complexes en trois types, en fonction de la forme générale de l'ARN : simple brin (ss), double brin (ds) et ARN de transfert mature (tRNA). Puis, nous avons modélisé une fonction de score pour chaque catégorie de complexes. Pour chaque fonction de score ainsi modélisée, nous avons procédé, en amont de l'apprentissage et de l'évaluation, à un filtre des complexes pour ne garder que les complexes de la catégorie correspondant à la fonction.

Les résultats de l'évaluation indiquent que les complexes mettant en jeu un tRNA mature sont un peu mieux modélisés avec une fonction de score dédiée (voir tableau 2.4). Les deux autres fonctions de score dédiées ne montrent pas d'amélioration par rapport à la fonction de score ROGER à poids positifs.

2.4.3 Combinaison linéaire à poids positifs

Après apprentissage de la fonction de score à combinaison linéaire par ROGER avec des coefficients des termes de score contraints aux valeurs dans [0 ; 1], la fonction de score est évaluée. Pour chaque mesure d'évaluation, nous allons évaluer les 120 fonctions de score générées par *leave-"one-pdb"-out* et la fonction de score globale sur le jeu de données issu des *Benchmarks* I et II.

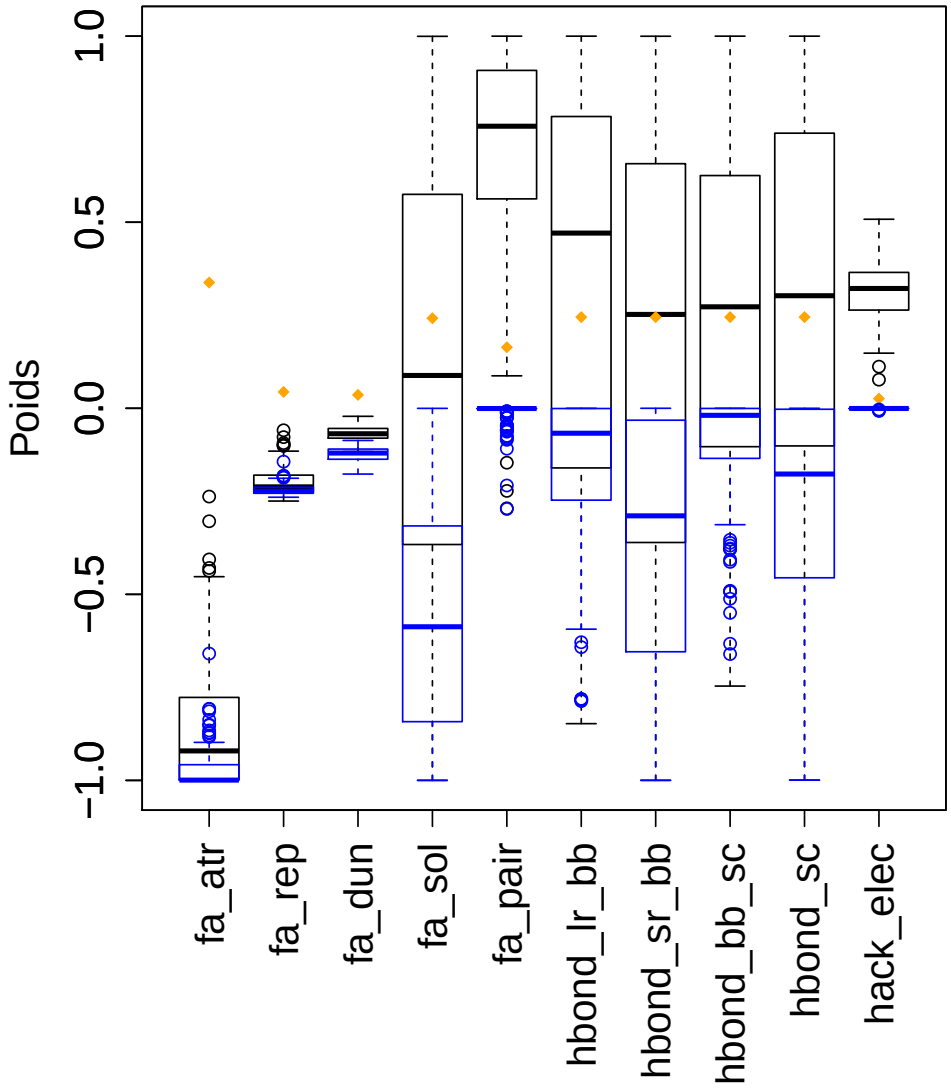


FIGURE 2.9 – Répartition de la valeur des coefficients par poids et contrainte d'intervalle de valeurs, pour les 120 fonctions de score générées par le *leave-"one-pdb"-out* mis en œuvre sur la PRIDB. Les points en losange jaune correspondent aux valeurs des coefficients des poids de la fonction de score par défaut de RosettaDock, ROS. Les boîtes à moustache représentent les valeurs des coefficients des termes de score pour un intervalle de définition négatif (en bleu) ou sans contrainte (en noir).

2.4.3.1 Pouvoir prédictif de la fonction de score

La ROC-AUC est supérieure à 0.7 pour 89 des 120 complexes, montrant un pouvoir prédictif important (voir tableau S12). Globalement, les presque-natifs sont donc bien séparés des leurres. La ROC-AUC est en moyenne de 0.8 et sa variance de 0.02.

Lges courbes ROC ont une pente à l'origine élevée, comme on peut le voir pour les

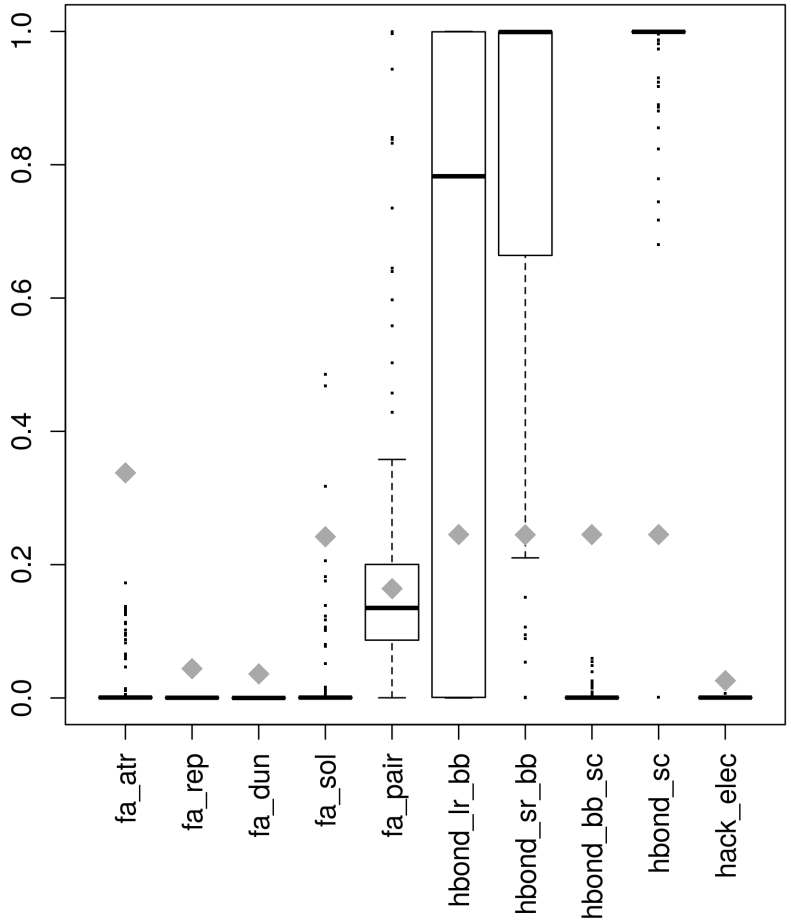


FIGURE 2.10 – Répartition de la valeur des coefficients par poids, pour les 120 fonctions de score générées par le *leave-"one-pdb"-out* mis en œuvre sur la PRIDB. Les points en losange correspondent aux valeurs des coefficients des poids de la fonction de score par défaut de RosettaDock, ROS.

complexes étant à la médiane, au premier quartile et au troisième quartile en ROC-AUC (voir fig. 2.11). Sur l'ensemble des complexes, seuls 8 des 120 ont une pente à l'origine insatisfaisante (voir fig. 2.12).

Dans le jeu de données issu des *Benchmarks* I et II, les résultats des fonctions de score ROGER sont similaires à ceux pour le jeu de données de perturbations issu de la PRIDB (voir fig 2.11).

Catégorie	Médiane(ROC-AUC)	ROC-AUC > 0.7	ROC-AUC > 0.6	Complexes
POS	0.82	0.73	0.88	120
POS ds	0.81	0.69	0.87	55
POS ss	0.80	0.76	0.88	33
POS tRNA	0.87	0.84	0.97	32

TABLE 2.4 – Évaluation du modèle de prédiction dédié par catégorie de complexes. Les catégories double brin (POS ds), simple brin (POS ss) et ARN de transfert (POS tRNA) sont comparées à la fonction de score sans catégorisation des complexes (POS). La médiane de la ROC-AUC, et la proportion des complexes vérifiant une condition sur la ROC-AUC parmi les complexes de la catégorie sont comparées. La dernière colonne correspond au nombre de complexes dans la catégorie. On peut remarquer que la fonction de score dédiée aux complexes avec un ARNt mature donne de meilleures performances que la fonction de score traitant indifféremment les complexes. Les deux autres fonctions de score dédiées donnent des résultats similaires à la fonction de score non dédiée.

2.4.3.2 Enrichissement du tri des candidats

Le score d'enrichissement défini au chapitre 1 montre un enrichissement du tri pour 27 des 120 complexes : ces 27 complexes ont un score d'enrichissement supérieur à 6. Sur les 120 complexes, 6 complexes ont un enrichissement inférieur à 1. À la lumière de ces résultats, il faut rappeler que la fonction de score apprend à séparer deux classes - presque-natifs et leurres - qui sont distinguées l'une de l'autre par un seuil fixe de 5 Å en IRMSD. Or, le score d'enrichissement considère les 10 % premiers candidats en IRMSD comme des presque-natifs et regarde parmi eux la proportion des 10 % premiers candidats en énergie. Dans un ensemble de candidats générés par faible perturbation, il arrive fréquemment qu'un seuil défini à 10 % des candidats en IRMSD soit très inférieur à 5 Å (de l'ordre de 1 à 2 Å). De manière générale, le score d'enrichissement est donc d'autant moins bon que le seuil à 10 % des candidats en IRMSD s'écarte de 5 Å. Le résultat obtenu est plus cohérent avec les autres mesures d'évaluation pour un score d'enrichissement à X % où X est la proportion de candidats nécessaires pour avoir un seuil en IRMSD de 5 Å. Pour ROGER, le score d'enrichissement ainsi calculé est supérieur à 6 pour 94 des 120 complexes, contre seulement 42 complexes pour *ROS*. Et aucun complexe n'a un score inférieur à 2 pour le score ROGER, contre 41 complexes pour *ROS*.

2.4.3.3 Détection d'entonnoir

Les courbes d'EvsIrms permettent de détecter un entonnoir pour 94 des 120 complexes (voir fig. S1). Ceci suggère que la fonction de score est utilisable en amarrage atomique pour affiner une structure 3D dont on a détecté l'épitope.

On pourrait penser que la séparation des candidats en presque-natifs et leurres selon un seuil fixe a un impact sur la signification de la prédiction. Si la fonction de

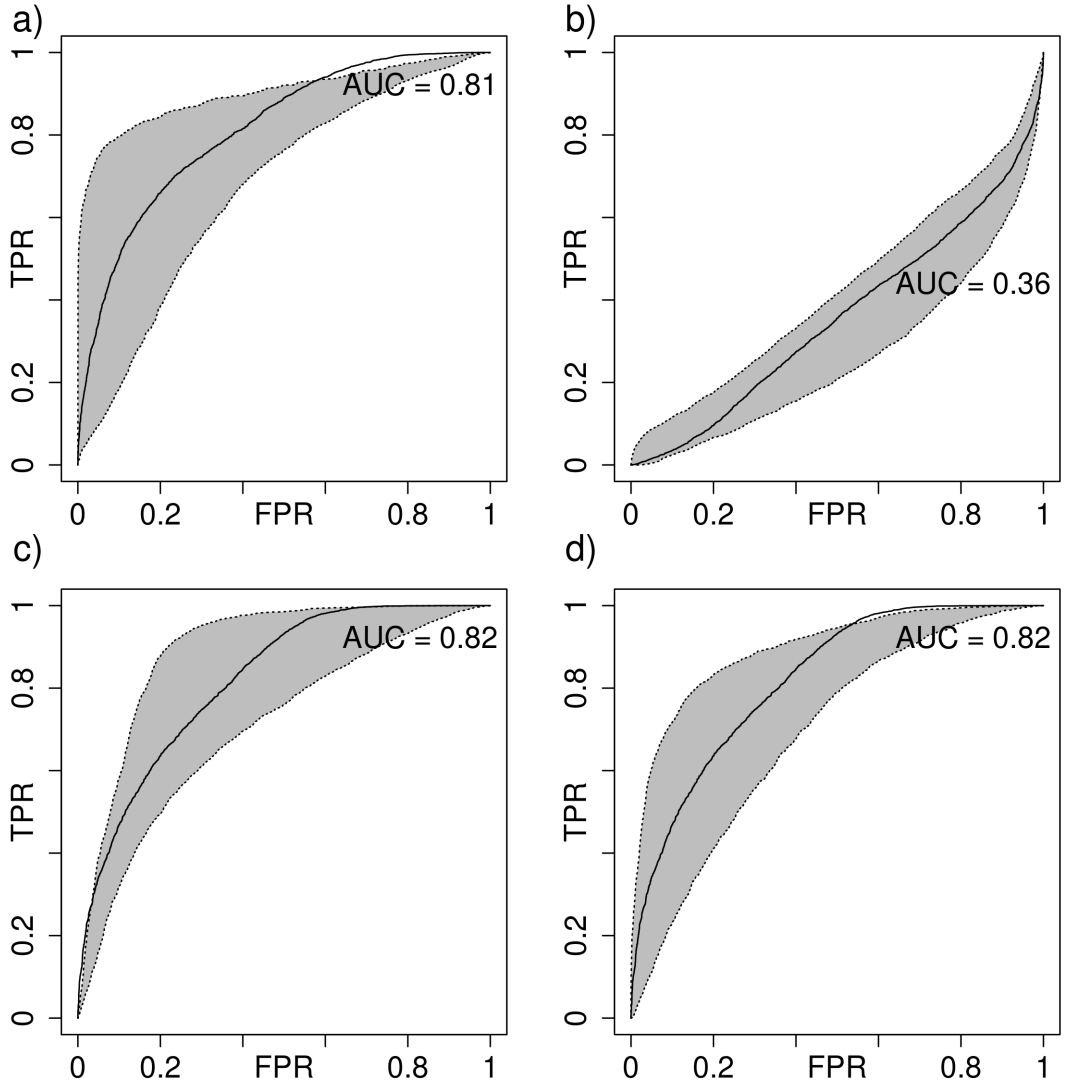


FIGURE 2.11 – Courbes ROC pour trois complexes de la PRIDB avec la fonction de score POS (a) et ROS (b), avec les courbes ROC pour trois complexes du *Benchmark* I (c) et du *Benchmark* II (d). Les trois complexes utilisés à chaque fois sont choisis parce qu'ils sont les plus proches de la médiane (en trait plein), le 1er quartile et le 3^{e} quartile (en pointillés) en ROC-AUC. La ROC-AUC de la médiane est à chaque fois indiquée.

score apprend correctement à séparer les classes, les candidats prédits presque-natifs ont une forte chance d'avoir un IRMSD $\leq$ 5 Å. Mais la prédiction pourrait ne pas donner d'information plus forte sur la divergence entre le presque-natif et la structure native. Pourtant, on constate avec la détection d'entonnoir que les caractéristiques de l'interaction ont correctement été modélisées par la fonction de score pour approximativement 100 des 120 complexes. En effet, les candidats de meilleure énergie tendent

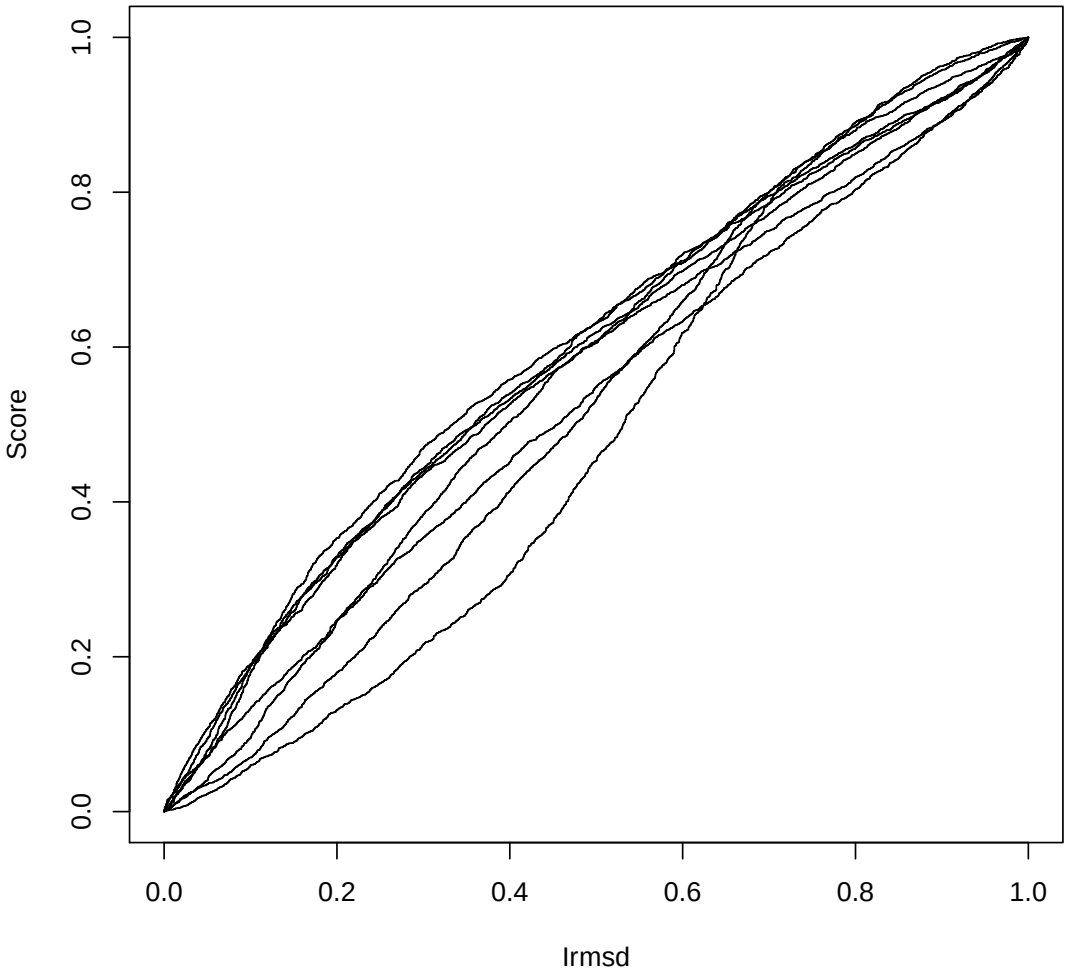


FIGURE 2.12 – Courbes ROC pour 8 complexes de la PRIDB avec la fonction de score POS. Ces 8 complexes ont été choisis pour la faible valeur de leur pente à l'origine.

aussi à être ceux qui ont le meilleur IRMSD et non n'importe quel IRMSD entre 0 et 5 Å. Cette information a été retrouvée par le modèle de fonction de score alors qu'elle n'était pas fournie en entrée de l'apprentissage.

Il reste néanmoins des cas où un entonnoir n'est tout simplement pas détecté. Pour ces complexes pour lesquels l'interaction n'est pas capturée, une structure 3D alternative est quelque fois proposée par la fonction de score, d'énergie plus élevée que l'interaction recherchée. Il s'agit ici de 1jbs et 1t0k, pour lesquels plusieurs leurres de même IRMSD sont proposés par la fonction de score.

Mais il y a aussi des cas où la fonction de score privilégie de façon consistante des structures plus éloignées de la native, sans pour autant qu'il s'agisse de leurres. C'est

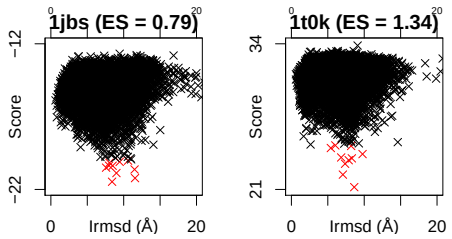


FIGURE 2.13 – Diagrammes d'énergie en fonction du IRMSD pour deux complexes de la PRIDB avec la fonction POS. On constate que la fonction de score propose une interaction différente de celle de la structure native, alors qu'elle détecte très bien un entonnoir pour 94 des 120 complexes.

notamment le cas pour 1feu, 1j1u ou 1pgl, ainsi que, dans une moindre mesure, pour 1asy, 1av6 ou 1b23. Pour certains de ces complexes, les candidats mis en avant par la prédiction évitent davantage l'interpénétration des partenaires que la structure native. Pour d'autres, un entonnoir est même visible en plus du premier entonnoir et laisse suspecter une potentielle interaction de plus haute énergie. Les complexes 2bx2, 2d6f et, surtout, 3egz montrent en effet qu'une seconde interaction est compatible avec le modèle de fonction de score (voir fig. 2.14).

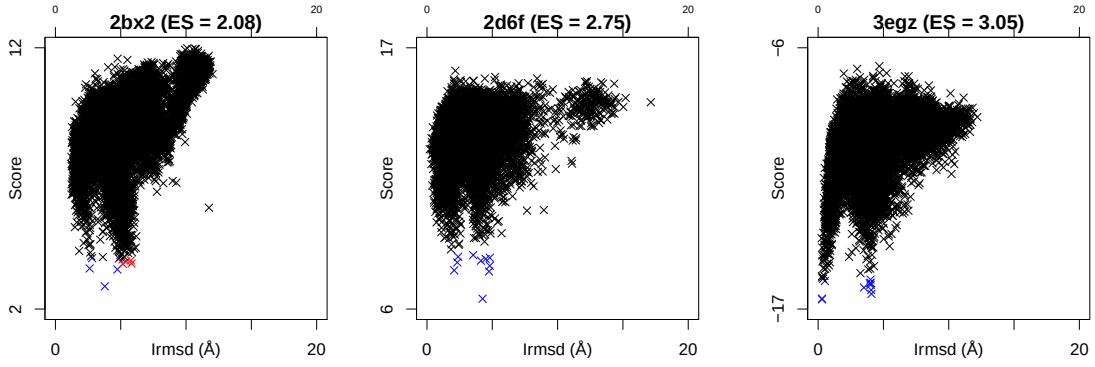


FIGURE 2.14 – Diagrammes d'énergie en fonction du IRMSD pour trois complexes de la PRIDB avec la fonction POS. On constate que la fonction de score propose, en plus de presque-natifs proches de la structure native, une interaction alternative à celle de la structure native.

2.4.3.4 Répartition des coefficients des termes de score

Les coefficients des termes de score nous permettent de mieux comprendre les forces en jeu influençant le plus les interactions entre protéines et ARN. On peut remarquer plusieurs types de comportements parmi tous les complexes. Notamment, le

comportement le plus remarqué est celui où les termes de score hbond, représentant la stabilité provenant des liaisons hydrogène, ont des coefficients élevés. Mais sur les quatre types de termes de score hbond, seuls trois d'entre eux ont des valeurs élevées : le coefficient reste très faible pour le score hbond_bb_sc, qui représente les liaisons hydrogène entre le squelette d'un partenaire et les chaînes latérales de l'autre partenaire. Ceci indique que les complexes protéine-ARN ont tendance à former des liaisons entre les atomes de leurs squelettes à courte et moyenne portée, mais aussi entre les atomes des chaînes latérales.

Code PDB	fa_atr	fa_rep	fa_dun	fa_sol	fa_pair
1h3e	0.000496	0.000793	0.000009	0.000944	0.22261
1j2b	0.000653	0.000337	0.000023	0.000474	0.175237
3ex7	0.00058	0.000314	0.000009	0.000511	0.166342
Code PDB	hbond_lr_bb	hbond_sr_bb	hbond_bb_sc	hbond_sc	hack_elec
1h3e	0.999557	0.999906	0.000132	0.999968	0.000726
1j2b	0.999245	0.999941	0.000627	0.99933	0.000559
3ex7	0.999739	0.999963	0.000301	0.999195	0.000479

TABLE 2.5 – Trois exemples types de coefficients des termes de score hbond élevés. Les valeurs des coefficients tournent toujours autour de valeurs très faibles ($< 10^{-3}$) ou très hautes ($> 1-10^{-3}$). Le seul terme de score faisant exception à la règle est le terme de score fa_pair, modélisant l'affinité entre atomes en fonction de leur environnement.

2.4.4 Filtrage *a priori* des candidats du modèle atomique

Avec le filtrage *a priori* des candidats pour chaque complexe, le modèle rejette d'emblée de la prédiction une partie des candidats, principalement pour deux raisons. D'une part, les candidats peuvent être rejetés parce qu'ils sont des leurres évidents. D'autre part, pour certains candidats, il peut être trop difficile de discriminer entre presque-natif et leurre. Il devient alors acceptable de rejeter ces candidats si la proportion de presque-natifs parmi eux est suffisamment faible et si les retirer du jeu d'apprentissage permet d'améliorer la prédiction.

Nous avons procédé à un filtre en fonction du score donné par le terme de score fa_rep, qui donne une quantification du chevauchement entre les deux partenaires. Nous ne conservons que les candidats en-dessous de la médiane du terme de score de fa_rep, ce qui signifie que nous choisissons systématiquement de ne conserver que les 5 000 meilleurs candidats en fa_rep. En réalité, il ne s'agit pas exactement des 5 000 meilleurs candidats en fa_rep. Le nombre exact de candidats retenu peut être supérieur à 5 000, puisque la valeur en fa_rep peut être identique entre candidats autour du seuil des 5 000 meilleurs candidats en fa_rep.

Sur les 120 complexes de la PRIDB, 54 n'ont que des complexes presque-natifs après utilisation du filtre. Ceci montre l'importance du terme de score fa_rep dans la

détermination des structures représentant le mieux l'interaction, mais aussi que ce terme de score ne suffit pas à prédire l'interaction.

Sur les 66 complexes restants, qui disposent encore de leurres, on constate une diminution drastique des performances en ROC-AUC. Sur ces complexes, 28 ont une diminution de plus de 0.05 du ROC-AUC pour POS. Le filtre a permis d'écarter des candidats qui étaient des leurres très faciles à discerner des presque-natifs pour les fonctions de score employées.

Cependant, nous observons aussi une augmentation du nombre de presque-natifs dans le top10. Les 3 complexes pour lesquels il est le plus difficile pour POS de prédire des presque-natifs dans le top10 ont, après filtre, tous au moins un presque-natif dans le top10 (voir tableau 2.6). De plus, le nombre de complexes pour lesquels il n'y a aucun presque-natif dans le top10 passe de 59 à 22 pour la fonction de score ROS.

pdb	Enrichissement		Top10		Top100		Presque-natifs	Seuil	ROC-AUC	
	ROS	POS	ROS	POS	ROS	POS			ROS	POS
1jbs	0.38	0.40	0	1	35	32	2528	5003	0.49	0.55
1t0k	0.61	1.03	9	8	79	76	3725	5002	0.51	0.56
2anr	0.38	1.04	8	6	43	75	2927	5000	0.43	0.56

TABLE 2.6 – Les trois complexes les plus difficiles à prédire pour la fonction de score POS ont tous au moins 1 candidat presque-natif dans le top10 après utilisation du filtre de la médiane en fa_rep, alors qu'ils étaient les seuls à avoir un nombre de presque-natif dans le top10 à 0 sans usage du filtre. Pourtant, le ROC-AUC reste faible (0.55 environ).

Nous pourrions être tentés de conserver le filtrage par la médiane en fa_rep dans la suite de ces travaux. En effet, ce filtre *a priori* permet d'améliorer les résultats quant à l'objectif fixé de maximiser le nombre de presque-natifs dans le top10. Cependant, il faut se rappeler que les candidats rejetés par ce filtre *a priori* sont rejetés parce qu'ils présentent une interpénétration trop importante. Or, de tels candidats ne sont pas attendus après une génération de candidats ayant incorporé l'utilisation d'un filtre adapté pour rejeter les candidats présentant des caractéristiques trop éloignées des attentes biologiques (interpénétration trop importante, distance trop importante entre les atomes des partenaires, *etc.*).

Il est ainsi conseillé d'employer le filtrage par la médiane en fa_rep en l'absence de tout autre filtre ou de tout amarrage pouvant garantir que les candidats générés ne présentent pas d'interpénétration n'ayant aucun sens biologique. Mais l'objectif dans les chapitres suivants est justement d'arriver à un protocole de prédiction d'interactions protéine-ARN où les candidats soumis à la prédiction atomique ont un sens biologique.

2.5 Conclusions sur la fonction de score atomique

Nous avons pu voir dans ce chapitre la fonction de score atomique implémentée par défaut dans RosettaDock, ROS. Vu que ROS n'est pas optimisé pour la prédiction d'interactions protéine-ARN, ses performances ne permettent pas la prédiction d'interactions entre protéine et ARN, encore moins le raffinement de structures 3D. Nous avons constaté que les mesures d'évaluation globales ne permettent pas d'obtenir d'informations sur l'objectif fixé, qui est de maximiser le nombre de presque-natifs dans le top10 des candidats. Dans cette optique, les mesures d'évaluation locales se sont avérées plus utiles. L'optimisation de la fonction de score atomique, en conservant les contraintes de RosettaDock, a permis d'obtenir des performances bien meilleures. Ce gain de performance a été conditionné à un apprentissage contraint à l'intervalle de définition positif, $[0\,;\,1]$, raison pour laquelle la fonction de score apprise s'appelle POS. Nous avons confronté POS à deux autres fonctions de score, ALL et NEG, respectivement sur l'intervalle de définition $[-1\,;\,1]$ pour ALL et $[-1\,;\,0]$ pour NEG. Cette confrontation a permis de montrer l'efficacité de POS, dont l'intervalle de définition est tiré de la connaissance *a priori* de l'amarrage protéine-protéine. Un filtre *a priori* permet d'améliorer la prédiction donnée par ROS, mais les résultats sont encore peu stables, marquant des écarts de performance importantes entre les différents complexes. Ce filtre consiste à ne conserver pour la prédiction que les candidats dont la valeur du terme de score fa_rep est inférieure à celle de la médiane de fa_rep sur les 10 000 candidats générés.

Nous avons donc obtenu une fonction de score POS non seulement capable de prédire l'interaction protéine-ARN, mais aussi d'être efficace dans le raffinement de structures 3D protéine-ARN. Il reste maintenant à concevoir une fonction de score s'affranchissant des contraintes de RosettaDock, pour espérer mieux modéliser encore l'interaction. La finalité est d'obtenir un protocole d'amarrage de prédiction d'interaction protéine-ARN multi-échelle, ce qui implique la modélisation d'une fonction de score à une autre échelle. Cette fonction de score sur une autre échelle peut servir de filtre *a priori* pour diminuer les temps de calcul et écarter les leurres les plus simples. C'est pourquoi, dans un second temps, nous nous focaliserons sur une échelle gros-grain.

Chapitre 3

Approche *a posteriori*

Nous avons vu, dans le chapitre précédent, l'approche classique mise en œuvre dans RosettaDock pour la génération et l'évaluation de candidats. Le choix de RosettaDock impose de se restreindre à une fonction d'évaluation qui est une combinaison linéaire des attributs associés à chaque candidat.

Comme nous le verrons par la suite, RosettaDock peut générer des candidats presque-natifs qui, d'après la fonction d'évaluation utilisée, ne sont pas classés parmi les meilleurs candidats. Il n'est donc pas toujours possible, en utilisant seulement une fonction de type combinaison linéaire, de mettre en avant les meilleurs candidats générés par RosettaDock. Nous nous sommes donc focalisé sur d'autres familles de fonctions afin de pouvoir ordonner plus efficacement les candidats générés par RosettaDock. Sachant que ces fonctions ne pourront être appliquées qu'en post-traitement de RosettaDock, c'est-à-dire sans interaction directe avec RosettaDock, nous appelerons les approches mises en œuvre pour obtenir ces nouvelles fonctions de tri : modèles *a posteriori*.

3.1 Modèles *a posteriori*

Les modèles de fonction de score *a posteriori* ne sont pas utilisés directement par RosettaDock et peuvent ainsi s'affranchir de la contrainte de la combinaison linéaire. Les candidats sont initialement générés par RosettaDock puis triés par une fonction de score appliquée en post traitement. Notons qu'un tri *a posteriori* ne permet pas d'orienter la génération des candidats durant l'étape de génération. Sachant qu'il est possible de fournir à RosettaDock un ensemble de candidats *a priori* peu divergeant des solutions recherchées, nous pouvons parfaitement envisager de coupler des phases de génération de candidats suivies de sélection des meilleurs candidats en appliquant les fonctions de tri que nous présenterons par la suite. Les meilleurs candidats pourront alors être utilisés comme conformations initiales pour l'itération suivante de RosettaDock.

De nombreuses approches peuvent être mises en œuvre pour apprendre des fonctions de score à partir des attributs disponibles pour chaque candidat. Nous avons étudié différentes approches reposant sur l'apprentissage de modèles : fonctions de

score de type combinaison linéaire étendue, arbres ou règles de décision, modèle bayésien naïf, *etc*.

3.1.1 Combinaison linéaire

La combinaison linéaire de termes physico-chimiques est la seule forme de fonction de score que RosettaDock peut nativement utiliser. L'équation 3.1 montre pour le candidat X la combinaison linéaire $f(X)$ des attributs x_i où w_i représentent les poids qui doivent être appris pour chaque attribut.

$$f(X) = \sum_{i=1}^{|A|} w_i x_i \tag{3.1}$$

Cette modélisation peut être étendue en intégrant la notion de *valeur de centrage* associée à chaque attribut.

Les valeurs de centrage doivent être apprises pour chaque attribut. Ces valeurs vont permettre de déterminer si la contribution au score de chaque attribut est linéaire ou non.

La nouvelle fonction de score obtenue, f_c, est définie par l'équation 3.2, où X représente le candidat, x_i la valeur du i^e attribut, w_i son poids et c_i sa valeur de centrage. Ainsi, la prise en considération des valeurs de centrage, permet d'obtenir, pour chaque attribut, un comportement linéaire par partie : croissant (resp. décroissant) jusqu'à un seuil c_i puis décroissante (resp. croissant) après le seuil. Notons que la contribution de x_i est nulle au point c_i.

$$f_c(X) = \sum_{i=1}^{|A|} w_i |x_i - c_i| \tag{3.2}$$

À l'image des différents noyaux utilisés pour les SVM, d'autres types de fonctions peuvent être prises en considération : polynomiales, quadratiques, gaussiennes, *etc*.

Notre principal objectif n'étant pas d'apprendre une fonction d'ordonnancement des candidats, mais plutôt de pouvoir identifier efficacement les candidats presque-natifs, nous nous sommes focalisés sur des approches permettant d'apprendre des modèles prédicitifs.

3.1.2 Approches dites "explicatives" : arbres et règles de décision

Les arbres et les règles de décision appartienent à la famille des modèles "explicatifs" ou "compréhensibles". En effet, lorsqu'un arbre de décision est appris sur des données, il est aisé de proposer à l'expert une représentation graphique des données. L'expert peut alors très facilement comprendre le modèle obtenu et se l'approprier. L'une des limitations de ces approches (arbres ou règles de décision) est la taille de l'arbre obtenu (ou la quantité de règles).

Pour illustration, voici un exemple simple d'arbre de décision (voir fig. 3.1). Sur trois attributs fa_dun, fa_pair et fa_rep, 50 exemples fictifs sont utilisés pour la prédiction : 25 presque-natifs et 25 leurres. La valeur de chacun des attributs est dans l'intervalle [0 ; 1] pour les 50 exemples. Voici le résultat de l'apprentissage d'un arbre décisionnel sur ces 50 exemples fictifs :

- fa_dun$\geq$ 0.43
  - fa_rep$\leq$ 0.72 (presque-natifs) : 13 presque-natifs et 5 leurres
  - fa_rep$>$ 0.72 (leurres) : 16 leurres et 4 presque-natifs
- fa_dun$<$ 0.43
  - fa_pair$<$ 0.14 (presque-natifs) : 2 presque-natifs
  - fa_pair$\geq$ 0.14
    - fa_rep$>$ 0.65 (leurres) : 3 leurres
    - fa_rep$\leq$ 0.65 (presque-natifs) : 6 presque-natifs et 1 leurre

FIGURE 3.1 – Exemple d'arbre de décision sur 50 exemples fictifs répartis en 25 presque-natifs et 25 leurres, avec 3 attributs : fa_dun, fa_pair et fa_rep.

Cet exemple fictif commence par séparer en deux les 50 exemples selon fa_dun, avec un seuil de 0.43. Les exemples avec fa_dun$\geq$ 0.43 sont ensuite séparés selon fa_rep, avec un seuil de 0.72, à la suite de quoi deux feuilles sont donc créées. La première feuille correspond aux exemples avec fa_dun$\geq$ 0.43 et fa_rep$\leq$ 0.72 et est peuplé de 13 presque-natifs et 5 leurres. En prédiction, comme cette feuille est majoritairement peuplée de presque-natifs, un exemple dont la classe est à prédire et qui serait attribué à cette feuille serait prédit presque-natif. La seconde feuille correspond aux exemples avec fa_rep$>$ 0.72 parmi les exemples de fa_dun$\geq$ 0.43. Cette feuille est peuplée de 16 leurres et 4 presque-natifs, et prédit comme un leurre tout exemple correspondant à ses seuils.

Pour les exemples avec fa_dun$<$ 0.43, une seconde séparation est effectuée avec un seuil de 0.14 sur fa_pair. Nous tombons directement sur une feuille pour fa_dun$<$ 0.43 et fa_pair$<$ 0.14, peuplée de 2 presque-natifs. Pour fa_dun$<$ 0.43 et fa_pair$\geq$ 0.14, une troisième séparation selon fa_rep est effectuée, avec un seuil de 0.65. Les exemples avec fa_dun$<$ 0.43, fa_pair$\geq$ 0.14 et fa_rep$>$ 0.65 sont au nombre de 3 et sont tous des leurres. La dernière feuille a 7 exemples, dont 6 presque-natifs et 1 leurre, avec fa_dun$<$ 0.43, fa_pair$\geq$ 0.14 et fa_rep$\leq$ 0.65. Cette feuille prédit les exemples comme étant des presque-natifs.

Dans cet exemple fictif, on peut remarquer deux cas de figure : les feuilles où les exemples sont de même classe et les feuilles où il y a des exemples des deux classes. Dans le cas où les exemples sont de même classe, il paraît évident que les exemples à prédire dont les valeurs d'attributs correspondent à cette feuille sont prédits comme étant de la classe des exemples de la feuille. Dans le second cas, c'est la classe majoritaire qui est considérée comme étant la classe de la feuille.

On peut aussi remarquer que les deux premières feuilles sont peuplées d'exemples des deux classes, alors qu'il reste un dernier attribut selon lequel les exemples n'ont pas été séparés. L'apprentissage de l'arbre a considéré que cette séparation en fa_pair n'apportait pas d'information supplémentaire à la prédiction.

La taille de l'arbre est ici relativement petite, mais cette taille peut grandement augmenter avec le nombre d'attributs et d'exemples.

Le principal intérêt des arbres (ou règles) de décision dans notre travail ne réside pas dans leur pouvoir explicatif mais plutôt dans la méthode mise en œuvre pour apprendre les modèles.

L'apprentissage des arbres de décision repose sur le principe *diviser-pour-régner*.

Le principe général est de diviser itérativement, chaque fois selon un attribut, l'ensemble des exemples en sous-ensembles pour lesquels les exemples ont plus de probabilité d'être d'une classe donnée. La construction de l'arbre de décision débute l'ensemble des exemples. L'objectif est de déterminer le couple (attribut, valeur) permettant d'obtenir une partition "optimale" des données. Plusieurs mesures permettent d'évaluer la qualité des partitions obtenues. Parmi les critères classiquement utilisés, nous pouvons citer le gain de Gini, l'entropie de Shannon, *etc*.

Sans rentrer dans les détails techniques de l'algorithme que nous avons utilisé, à savoir C4.5 [207] (algorithme de référence pour les arbres de décision), nous précisons tout de même que le critère utilisé pour sélectionner les couples (attribut, valeur) permettant de construire un arbre de décision est l'entropie de Shannon.

L'algorithme ne procède pas à une séparation en nœuds fils s'il se trouve dans l'un des cas suivants :

- tous les exemples du nœud courant sont de la même classe y, ce qui crée une feuille étiquetée avec la classe y ;
- tous les attributs obtiennent un gain d'information normalisé nul, ce qui crée une feuille étiquetée de la classe majoritaire ;
- la taille d'au moins une des partitions créé est inférieure à un seuil prédéfini.

Dans le premier cas, le classement est parfait suivant les exemples du jeu d'apprentissage. Dans le second cas, les attributs ne permettent pas d'améliorer davantage le gain d'information et toute inférence supplémentaire de règles de décision est donc inutile du point de vue des données disponibles en apprentissage. Le troisième cas correspond à la situation où, malgré l'ensemble des tests déjà effectués, la partition courant n'est toujours pas pure mais sa taille est telle qu'il n'est pas raisonnable de continuer à rafiner car cela impliquerait du surapprentissage.

D'autres arbres de décision peuvent être obtenus, par exemple des arbres de décision partiels tels que l'algorithme PART permet d'en construire. L'algorithm PART génère des arbres de décision partiels sans optimisation globale des règles apprises. La même stratégie d'apprentissage que celle présentée précédemment pour les arbres de décision, à savoir diviser-pour-régner, est utilisée dans PART. PART commence par créer un arbre de décision élagué, qui divise les candidats en sous-ensembles dans les nœuds de l'arbre. Pour ce faire, PART divise en sous-ensembles les candidats d'un sous-ensemble n'étant pas encore divisé selon un test sur la valeur d'un attribut au hasard. PART répète ce processus récursivement jusqu'à ce que tous les sous-ensembles soient des feuilles contenant des candidats d'une seule classe. Pour l'élagage de l'arbre, PART annule la division de n'importe quel sous-ensemble en feuilles si l'erreur calculée sur les feuilles est supérieure ou égale à l'erreur calculée sur le sous-ensemble. Une fois un arbre de décision élagué, PART infère une règle de décision en ne choisissant que le chemin de la racine vers la feuille la plus peuplée

de l'arbre. Un nouvel arbre de décision est ensuite construit en ne considérant que les candidats qui ne sont pas couverts par une règle de décision. PART itère jusqu'à couvrir tous les candidats.

Comme indiqué précédemment, parmi les approches dites "explicatives", nous trouvons des approches permettant d'apprendre des règles de décision. Parmi ces approches, nous avons retenu RIPPER qui est une amélioration de IREP *(Incremental Reduced Error Pruning)* [50]. L'apprentissage des règles de décision avec RIPPER repose sur deux étapes s'exécutant séquentiellement : 1) la construction avec croissance et élagage et 2) l'optimisation des règles de décision. Pour chaque classe, de la moins peuplée à la plus peuplée, RIPPER commence par itérer l'étape de construction sur l'ensemble des attributs par séries de croissance et d'élagage de règles. Une étape de construction commence par une partition de l'ensemble d'apprentissage en un ensemble de croissance et un ensemble d'élagage. La construction des règles s'arrête lorsque :

- la longueur de description de la règle en construction a 64 bits de plus que celle de la règle de longueur de description la plus petite parmi les règles trouvées ;
- il n'y a plus d'exemple positif, auquel cas l'ensemble de règles est retourné tel quel ;
- le taux d'erreur est supérieur ou égal à 50 %, auquel cas l'ensemble de règles retourné contient toutes les règles construites sauf la dernière règle.

La croissance rajoute une à une les conditions permettant d'augmenter le plus vite le gain d'information (par approche gloutonne), jusqu'à obtenir une précision de 100%. La croissance teste toutes les valeurs possibles à chaque fois qu'elle ajoute une condition. L'élagage est incrémental pour chaque règle : l'élagage supprime la condition dont l'effacement améliore la mesure d'élagage. La mesure d'élagage utilisée dans JRip (implantation de RIPPER dans Weka) est égale à $(p+1)/(p+n+2)$ avec p le nombre d'exemples positifs correctement prédits et n le nombre d'exemples négatifs correctement prédits parmi les exemples d'élagage. Lorsqu'une règle est construite, les exemples qu'elle couvre sont retirés des exemples utilisés pour la construction des futures règles.

L'étape d'optimisation consiste en la construction de deux variantes pour chaque règle, en utilisant comme ensemble d'apprentissage un sous-ensemble aléatoire des exemples de l'ensemble d'apprentissage. La première variante ajoute des conditions à une règle vide. La seconde variante ajoute les conditions par approche gloutonne à la règle initiale. Pour l'optimisation, le mesure d'élagage est $(p+n)/(P+N)$, avec P le nombre d'exemples positifs et N le nombre d'exemples négatifs parmi les exemples d'élagage. Une fois les variantes construites, la variante de longueur de description la plus petite est conservée. Enfin, les règles augmentant la longueur de description de l'ensemble de règles sont retirées.

3.1.3 Approches dites non explicatives

Les forêts aléatoires se fondent sur l'apprentissage de B arbres décisionnels d'une hauteur maximale h en utilisant à chaque fois k attributs au hasard. Pour chaque

arbre décisionnel (ou arbre aléatoire), un échantillon aléatoire du jeu de données d'apprentissage est utilisé pour l'apprentissage. De plus, pour chaque arbre décisionnel, seul un sous-ensemble de k attributs aléatoires est utilisé pour l'apprentissage. Un arbre décisionnel est l'arborescence d'une succession de tests conditionnels à effectuer sur les valeurs des attributs de l'exemple à prédire. À chaque nœud d'un arbre décisionnel correspond une test sur un attribut, menant à l'un de ses nœuds fils en fonction de la valeur de l'attribut testé. Pour décider de la classe d'un exemple en utilisant un arbre décisionnel, il faut successivement tester, du noeud racine jusqu'à une feuille, les nœuds vers lesquels dirigent le résultat de chaque decision. Un arbre décisionnel est appris par mesure statistique sur les différents attributs au sein des exemples de chacune des classes. Cette mesure statistique se fonde sur une approche fréquentiste.

Le classifieur bayésien naïf repose sur le théorème de Bayes. Le classifieur bayésien naïf est appelé naïf en ce sens qu'il fait l'hypothèse d'indépendance des différents attributs entre eux (voir éq. 3.3). La probabilité qu'un exemple soit d'une certaine classe y sachant la valeur de chacun de ses attributs est égale à sa probabilité *a priori* d'être de la classe y multipliée par le produit des probabilités qu'un exemple de cette classe ait la valeur de chacun de ses attributs. Les différents paramètres du modèle sont estimés par calcul de fréquences sur l'ensemble d'apprentissage. Il est par ailleurs fait l'hypothèse que les variables aléatoires sous-jacentes aux paramètres suivent une loi normale.

$$P(y|X) = P(y) \prod_{i=1}^{|A|} P(x_i|y) \tag{3.3}$$

Un autre classifieur est dit naïf, le 1-plus-proche-voisin. **Le 1-plus-proche-voisin** est une instance du k-plus-proche-voisin pour $k = 1$. Le k-plus-proche-voisin prédit qu'un exemple est de la classe majoritaire parmi les k exemples appris les plus proches. Pour k pair, en cas d'égalité, il est possible de privilégier une classe par rapport à l'autre ou de pondérer le vote de chaque voisin par l'inverse de sa distance à l'exemple à prédire Le choix de la distance employée influence la prédiction. Traditionnellement, la distance euclidienne est mesurée pour déterminer la distance entre deux exemples. Mais d'autres distances – comme la distance de Manhattan ou la distance de Mahalanobis – sont plus adaptées à certaines problématiques. Les coordonnées des exemples sont les valeurs de leurs attributs. Plus k est grand et moins la méthode est sensible aux erreurs présentes dans les jeux de données, mais aussi plus la limite entre les classes est floue. Le 1-plus-proche-voisin fait l'hypothèse que les exemples se comportent localement de la même manière : leur classe est la même si les valeurs de leurs attributs sont proches. De plus, tous les attributs sont comparables dans le poids qui leur est donné pour le calcul de la distance.

Tous ces classifieurs sont plus adaptés à certaines problématiques qu'à d'autres. Mais il existe des classifieurs dont la particularité est de tirer partir de la combinaison de classifieurs d'approches très différentes : il s'agit des **métaclassifieurs**. Si plusieurs classifieurs peuvent chacun capturer des informations différentes sur la prédiction, le métaclassifieur a plus de chances de donner une meilleure prédiction que chacun

des classifieurs dont il dépend. Les métaclassifieurs offrent le plus de flexibilité en combinant les scores de plusieurs modèles de fonction de score.

AdaBoost est un métaclassifieur reposant sur l'apprentissage successif de classifieurs faibles. Chaque itération d'AdaBoost apprend un nouveau classifieur faible. Pour le premier classifieur appris, tous les exemples ont un poids identique. Chaque autre classifieur faible appris reçoit en entrée de son apprentissage, pour chaque candidat, un poids augmentant avec les erreurs commises par les classifieurs faibles précédents sur ce candidat. Cette pondération donne plus d'importance aux exemples n'ayant pas été capturés par les classifieurs faibles précédemment construits. Au fur et à mesure des itérations, les exemples les plus difficiles à classer reçoivent un poids élevé, jusqu'à ce que l'un des classifieurs les classe correctement.

Vote attribue pour chaque exemple à prédire une classe selon un principe de vote. Le principe de vote peut utiliser la moyenne, la médiane, le minimum ou même le maximum.

Les machines à vecteurs de support (SVM, *Support Vector Machines*) ont pour principe général d'apprendre une fonction de séparation qui maximise la marge entre les exemples et leur séparation. Maximiser la marge de séparation a pour objectif de minimiser l'erreur de généralisation. Les SVM mettent en jeu des noyaux pour déterminer cette séparation. Avec les SVM, le modèle est paramétré par l'hyperplan utilisé pour séparer les exemples prédits positifs des exemples prédits négatifs. De plus, il est possible d'utiliser une fonction noyau pour transformer les exemples et générer un modèle implicitement non-linéaire. Selon la problématique, les différents noyaux sont plus ou moins adaptés. Les SVM ne tiennent cependant pas compte de la différence de distribution des exemples en fonction de leur classe. En effet, les SVM ont pour objectif de maximiser la marge de séparation, *i.e.* la distance minimale entre la fonction de séparation apprise et les exemples de chaque classe. Or, les exemples peuvent se distribuer différemment selon la classe à laquelle ils appartiennent. Les exemples d'une classe peuvent être très proches les uns des autres alors que les exemples d'une autre classe peuvent être comparativement plus étendus. Dans un tel cas de figure, il serait plus judicieux, pour minimiser l'erreur de généralisation, de donner un poids plus important à la distance aux exemples de la classe la plus étendue qu'aux autres exemples. La fonction de séparation serait alors plus proches des exemples de la classe dont la distribution est la moins étendue.

3.1.4 Boîte à outils utilisée pour apprendre les modèles

Weka [103] est une boîte à outils implémentée en Java d'algorithmes de fouille de données. Weka prend en données d'entrée des exemples munis de descripteurs et peut visualiser la corrélation entre ces descripteurs au sein des données. Weka peut filtrer les données d'entrée et utiliser des méthodes de clustering ou de sélection de descripteurs. Weka peut aussi apprendre des modèles de prédiction sur ces données, les tester et sortir les modèles et leurs évaluations. Les classifieurs disponibles dans Weka sont rangés par catégories : les classifieurs bayésiens, les fonctions, les classifieurs naïfs, les métaclassifieurs, les règles de décision, les arbres de décision, les classi-

fieurs multi-instances et les classifieurs n'ayant pas pu être rangés dans les autres catégories. Les classifieurs sont appris avec la version 3.6 de Weka. Les classifieurs présents dans Weka et utilisés ici sont :

- J48 (implémentation de C4.5 [207]) ;
- JRip (implémentation de RIPPER, *Repeated Incremental Pruning to Produce Error Reduction* [50]) ;
- PART (variante de C4.5 et de RIPPER reposant sur des arbres de décision partiels [86]) ;
- RandomForest (implémentation des forêts aléatoires [30]) ;
- NaiveBayes (implémentation du classifieur bayésien naïf [127]) ;
- IB1 (implémentation du 1-plus-proche-voisin [2]) ;
- AdaBoostM1 (implémentation d'AdaBoost [223]) ;
- Vote [135] ;
- Machines à vecteurs de support (SVM [38]).

3.1.5 Méthodologie d'optimisation des fonctions de score *a posteriori*

La méthodologie générale d'optimisation des fonctions de score *a posteriori* se décompose en quatre phases :

- reprendre les jeux de données préparés et présentés au chapitre 1 ;
- l'apprentissage des fonctions de score par les différents classifieurs sur l'ensemble des 120 complexes et le même apprentissage, mais effectué en *leave-"one-pdb"-out* ;
- l'apprentissage par ROGER des métascores avec comme attributs les termes de score physico-chimiques et les scores des différents classifieurs ;
- l'évaluation des différentes fonctions de score des classifieurs ainsi que des métascores.

3.2 Évaluation des fonctions de score *a posteriori*

A posteriori, il est possible d'utiliser une fonction de score plus complexe qu'une combinaison linéaire de termes d'énergie. L'utilité de ce procédé est de pouvoir retrier un ensemble de candidats générés par RosettaDock, pour pouvoir réinjecter les meilleurs candidats dans un amarrage atomique, pour affiner les structures obtenues. Cela nécessite que la fonction de score *a posteriori* permette de mieux identifier les presque-natifs qu'une fonction de score étant une combinaison linéaire de termes d'énergie. Les résultats suivants évaluent plusieurs modèles de fonctions de score *a posteriori*.

Pour que chaque terme de score puisse participer dans la même mesure à la fonction de score, il est nécessaire que l'intervalle de valeurs de chaque terme de score soit réduit à l'intervalle unité. Les termes de score à valeurs positives sont ramenés à l'intervalle [0 ; 1] alors que les termes de score à valeurs négatives sont ramenés à

l'intervalle $[-1\ ;\ 0]$. La répartition des candidats pour chaque terme de score est donc étudiée pour mieux comprendre l'influence de chaque terme.

3.2.1 Répartition des candidats par terme de score

Pour étudier au mieux le comportement des termes de score pour l'ensemble des candidats presque-natifs et leurres, les valeurs réduites à l'intervalle unité sont indiquées entre parenthèses, sauf pour les scores issus des classifieurs, qui donnent déjà des valeurs dans l'intervalle unité. L'étude se fait sur l'ensemble des 1 200 000 candidats et sur les termes de score physico-chimiques comme sur ceux issus des classifieurs (voir tableau 3.1).

On peut notamment remarquer que la valeur du terme de score fa_dun est plus élevée pour les presque-natifs que pour les leurres : la médiane est de 683 (0.10) pour les presque-natifs contre 549 (0.08) pour les leurres. La terme de score fa_dun correspond à la pénalité donnée par la base de données Dunbrack pour les rotamères présents dans la structure évaluée. Pour les deux classes d'exemples, les valeurs se distribuent densément autour de la médiane et dépassent largement la valeur moyenne, respectivement à 848 (0.12) et 682 (0.10). Cette répartition montre que seul un petit nombre d'acides aminés est, pour chaque complexe, assimilé à une pénalité élevée, mais aussi que ce nombre est plus élevé pour les presque-natifs que pour les leurres. On peut en conclure que les chaînes latérales des acides aminés se comportent différemment de ce qui est attendu par la base de données Dunbrack lorsqu'elles sont en interaction avec un ARN.

Les termes de score fa_rep et fa_pair sont quant à eux plus faibles pour les presque-natifs que pour les leurres. Pour fa_rep, les presque-natifs ont une médiane de 352 (0.27), alors que les leurres ont une médiane de 406 (0.32). Pour fa_pair, la médiane est à 37 (0.17), contre 51 (0.23) pour les leurres. Ils ont effectivement pour but de donner une pénalité aux candidats mettant en jeu des atomes trop proches les uns des autres (fa_rep) ou rarement rencontrés ensemble (fa_pair).

Le terme de score fa_sol a un comportement différent dans les valeurs négatives, puisqu'il a des valeurs plus faibles en valeur absolue pour les presque-natifs que pour les leurres. La médiane de fa_sol est de -9 (-0.22) pour les presque-natifs et de -11 (-0.27) pour les leurres.

Le score le plus remarquable issu des classifieurs est le score donné par le 1-plus-proche-voisin : sa moyenne est de 0.62 pour les presque-natifs et 0.55 pour les leurres.

Les autres scores appris à l'aide des classifieurs ne permettent pas de discriminer les presque-natifs des leurres. Mais nous allons tout de même évaluer l'ensemble des classifieurs à l'aide des différentes mesures d'évaluation.

3.2.2 Weka

Les fonctions de score générées avec les différentes stratégies d'apprentissage sélectionnées dans Weka montrent des performances hasardeuses (voir tableau 3.2).

Exemples	Score	1er quartile	Médiane	Moyenne	3e quartile
+1	fa_dun	366 (0.05)	683 (0.10)	848 (0.12)	1165 (0.16)
+1	fa_pair	19 (0.08)	37 (0.17)	50 (0.22)	70 (0.31)
+1	fa_rep	228 (0.18)	352 (0.27)	425 (0.33)	607 (0.47)
+1	fa_sol	-3 (-0.08)	-9 (-0.22)	-11 (-0.27)	-17 (-0.40)
+1	hack_elec	-8 (-0.09)	-17 (-0.21)	-18 (-0.23)	-25 (-0.31)
+1	hbond_bb_sc	-13 (-0.07)	-29 (-0.16)	-37 (-0.21)	-55 (-0.31)
+1	NearestNeighbor	0.00	1.00	0.62	1.00
+1	NaiveBayes	0.34	0.42	0.50	0.60
+1	JRip	0.44	0.54	0.51	0.57
+1	J48	0.44	0.52	0.53	0.74
+1	PART	0.45	0.52	0.55	0.68
+1	RandomForest	0.40	0.60	0.56	0.70
-1	fa_dun	282 (0.04)	549 (0.08)	682 (0.10)	927 (0.13)
-1	fa_pair	25 (0.11)	51 (0.23)	60 (0.27)	82 (0.37)
-1	fa_rep	267 (0.21)	406 (0.32)	487 (0.38)	671 (0.52)
-1	fa_sol	-6 (-0.13)	-11 (-0.27)	-14 (-0.32)	-20 (-0.48)
-1	hack_elec	-9 (-0.11)	-18 (-0.22)	-20 (-0.25)	-26 (-0.33)
-1	hbond_bb_sc	-17 (-0.10)	-40 (-0.23)	-46 (-0.26)	-60 (-0.34)
-1	NearestNeighbor	0.00	1.00	0.55	1.00
-1	NaiveBayes	0.34	0.41	0.47	0.54
-1	JRip	0.43	0.53	0.49	0.57
-1	J48	0.43	0.47	0.51	0.64
-1	PART	0.44	0.48	0.51	0.58
-1	RandomForest	0.40	0.50	0.53	0.70

TABLE 3.1 – Pour chaque terme de score sont indiqués la valeur du premier quartile, de la médiane, de la moyenne et du troisième quartile, sur les presque-natifs (exemples +1) et les leurres (exemples -1). Entre parenthèses se trouvent les valeurs des scores ramenées à l'intervalle unité quand ce n'est pas déjà le cas. NearestNeighbor correspond au score donné par la fonction de score apprise au moyen du 1-plus-proche-voisin, NaiveBayes au classifieur bayésien naïf, JRip à l'implémentation de RIPPER, J48 à l'implémentation de C4.5, PART est une variante de C4.5 et RandomForest à l'apprentissage de forêts aléatoires.

De manière générale, moins de la moitié des complexes ont une ROC-AUC supérieure à 0.5. Les performances de Vote ne sont pas indiquées car la fonction de score apprise par Vote classe systématiquement les candidats comme presque-natifs. Seul le classifieur bayésien naïf donne des résultats satisfaisants, avec 108 des 120 complexes ayant une ROC-AUC supérieure à 0.5 et 81 complexes avec un #presque-natifs (top10_E(Candidats)) non nul.

Malgré les performances, il est tout de même intéressant d'observer par exemple les premiers termes de score utilisés par les arbres de décision (voir fig. 3.2). Ce

Classifieur	ROC-AUC	ROC-AUC > 0.5	#(top10$_E$(Candidats))> 0
ROGER linéaire	0.798±0.018	119	117
JRip	0.314±0.164	36	8
IB1	0.449±0.122	53	3
PART	0.472±0.039	49	18
J48	0.493±0.020	60	16
RandomForest	0.498±0.022	60	56
NaiveBayes	0.649±0.015	108	81

TABLE 3.2 – Pour chaque classifieur Weka sélectionné sont indiqués : la moyenne et la variance de la ROC-AUC, le nombre de complexes pour lesquels la ROC-AUC est supérieure à 0.5 et le nombre de complexes avec un #presque-natifs (top10$_E$(Candidats)) non nul.

sont les termes de score fa_dun – qui correspond au terme de score des rotamères Dunbrack – hack_elec (terme de score électrostatique), fa_sol (terme de solvatation), fa_pair (terme d'affinité entre pairs d'atomes), fa_rep (terme de répulsion universelle) et les termes de score hbond qui sont mis en avant. Nous avons vu dans l'étude de la répartition des valeurs des termes de score entre presque-natifs et leurres que fa_dun donnait pour les presque-natifs des valeurs plus élevées que pour les leurres. Et en effet, une petite partie des presque-natifs correspond à un pic de valeurs extrêmes en fa_dun, avec de nombreuses chaînes latérales dans des conformations ordinairement peu probables dans une protéine. Ceci nous conforte dans l'idée que la distribution des conformations des chaînes latérales des acides aminés d'une protéine peut changer à l'interaction avec un ARN. Les termes de score électrostatique, d'affinité entre pairs d'atomes et des liaisons hydrogène ont déjà été vus dans la section 2.4.3.4 comme étant des facteurs importants de l'interaction protéine-ARN. Le terme de score fa_rep a aussi été vu dans le filtre par la médiane en fa_rep comme étant important pour écarter les leurres comportant trop d'interpénétration (voir section 2.4.4). Seul le terme de solvatation n'avait jusqu'à lors pas été vu dans le chapitre précédent comme important pour l'interaction protéine-ARN.

3.2.3 ROGER non linéaire

Lever la contrainte de linéarité de la fonction de score permet d'explorer des solutions plus complexes, pouvant prendre en compte des informations qu'une combinaison linéaire des termes de score n'atteint pas. L'apprentissage de fonctions de score non linéaires avec ROGER a cet objectif. Deux types de fonctions de score sont apprises : l'une apprise uniquement sur les termes de score physico-chimiques, l'autre apprise aussi sur les scores des classifieurs sélectionnés dans Weka.

Même si les classifieurs sélectionnés dans Weka ne donnent pas de résultats satisfaisants, ils ont des performances de classifieur faible pour plusieurs dizaines de complexes chacun. S'ils capturent chacun des informations différentes sur les candidats, il

- fa_dun $\leq$ 866.949
  - fa_dun $\leq$ 231.186
    - fa_sol ≤ -23.739
      - fa_atr ≤ -2023.861 (presque-natifs)
      - fa_atr > -2023.861 (leurres)
    - fa_sol > -23.739
      - hbond_sc ≤ -12.138 (leurres)
      - hbond_sc > -12.138 (presque-natifs)
  - fa_dun $>$ 231.186
    - fa_sol ≤ -35.702
      - hbond_bb_sc ≤ -93.066 (leurres)
      - hbond_bb_sc > -93.066 (presque-natifs)
    - fa_sol > -35.702
      - fa_rep $\leq$ 1042.462 (leurres)
      - fa_rep $>$ 1042.462 (presque-natifs)
- fa_dun $>$ 866.949
  - fa_dun $\leq$ 1499.729
    - hbond_bb_sc ≤ -148.009
      - fa_rep $\leq$ 1063.695 (leurres)
      - fa_rep $>$ 1063.695 (presque-natifs)
    - hbond_bb_sc > -148.009
      - hack_elec ≤ -2.433 (leurres)
      - hack_elec > -2.433 (presque-natifs)
  - fa_dun $>$ 1499.729
    - fa_pair $\leq$ 65.641
      - fa_rep $\leq$ 556.425 (leurres)
      - fa_rep $>$ 556.425 (presque-natifs)
    - fa_pair $>$ 65.641
      - hbond_sr_bb ≤ -8.178 (presque-natifs)
      - hbond_sr_bb > -8.178 (leurres)

FIGURE 3.2 – Arbre de décision appris avec J48 sur les 120 complexes de la PRIDB. Seuls les nœuds après au plus 4 bifurcations sont affichés, pour donner un aperçu des premiers attributs mis en jeu dans l'arbre.

peut être intéressant de combiner les forces de chacun de ces classifieurs. L'apprentissage d'un métascore avec ROGER non linéaire permet de tester cette hypothèse. Les scores des classifieurs sélectionnés dans Weka deviennent des termes de score, au même titre que les termes physico-chimiques.

Dans l'ensemble, les fonctions de score apprises avec ROGER non linéaire donnent des résultats moins satisfaisants que celles obtenues avec ROGER linéaire (voir tableau 3.3). Sans métascore, les résultats sont quasiment identiques concernant le nombre de complexes avec une ROC-AUC supérieure à 0.5, mais moins satisfaisants du point de vue des autres mesures d'évaluation. L'emploi des scores des classifieurs de Weka dans les métascores dégradent les performances. Les métascores donnent

tout de même 114 complexes pour lesquels il y a au moins un presque-natif dans le top10 des candidats triés en énergie, contre 106 pour ROGER non linéaire sans apprentissage d'un métascore.

Classifieur	ROC-AUC	ROC-AUC > 0.5	top10> 0
ROGER linéaire	0.798 ± 0.018	119	117
Métascore ROGER non linéaire	0.648 ± 0.015	107	114
ROGER non linéaire	0.649 ± 0.015	115	106

TABLE 3.3 – Pour chaque classifieur ROGER sélectionné sont indiqués : la moyenne et la variance de la ROC-AUC, le nombre de complexes pour lesquels la ROC-AUC est supérieure à 0.5 et le nombre de complexes avec un #presque-natifs (top10_E(Candidats)) non nul.

3.3 Conclusions sur la fonction de score *a posteriori*

Nous avons commencé par observer la répartition des termes de score en comparant les presque-natifs avec les leurres. Nous avons formulé plusieurs constatations, notamment que la base de données de rotamères Dunbrack n'est peut-être pas adaptée à la prédiction d'interaction protéine-ARN. En plus d'utiliser une base de données de rotamères pour les ARN, il peut être nécessaire d'utiliser une base de données spécifique aux interactions protéine-ARN pour obtenir les probabilités associées à chaque rotamère en interaction avec un ARN. Nous avons évalué des fonctions de score modélisées pour être utilisées *a posteriori* de la génération de candidats par RosettaDock. Nous avons modélisé des fonctions de score directement apprises grâce à des classifieurs usuels, notamment des arbres et règles de décision, mais aussi des métascores prenant pour attributs les scores donnés par ces classifieurs en plus des termes de score physico-chimiques. Il s'avère que les fonctions de score testées n'ont pas de performances plus intéressantes que celles de la fonction de score atomique POS.

C'est la raison pour laquelle nous nous tournons maintenant vers l'usage de termes de score non pas physico-chimiques, mais géométriques. Nous souhaitons étendre le protocole de prédiction des interactions protéine-ARN en testant la fonction de score atomique POS sur des candidats issus d'une prédiction aveugle. Or, POS n'a été testée que pour des candidats à au plus une quinzaine d'Angströms de la structure native. Nous souhaitons donc concevoir une fonction de score à une échelle plus gros-grain avec pour objectif de trouver des candidats utilisables par POS.

Chapitre 4

Approche multi-échelle

4.1 Principe de l'approche multi-échelle

Pour prédire l'interaction avec un maximum d'efficacité – temps de calcul le plus faible possible et qualité des prédictions – plusieurs approches peuvent être combinées. Nous avons vu l'utilisation de filtres *a posteriori*, pour réordonner les résultats du tri après une génération de candidats qui contiennent au moins quelques presque-natifs. Nous pouvons aussi intervenir en amont de la génération des candidats à l'échelle atomique, en générant des candidats à l'échelle gros-grain. Cette génération à l'échelle gros-grain a l'intérêt de possiblement donner un aperçu du score qu'aurait le même candidat à l'échelle atomique, mais avec un temps de calcul réduit.

4.1.1 Représentation géométrique gros-grain des acides aminés et des acides nucléiques

Le prix Nobel de Chimie 2013[7] a montré l'importance dans le domaine de la biologie structurale computationnelle des représentations multi-échelle combinées aux modèles physiques simples. Ce modèle correspond en effet au développement des premiers potentiels énergétiques des années 70. La première apparition correspond au modèle de Levitt [151] dans lequel chaque résidu d'une protéine était représenté par trois points encore appelés "atomes gros-grains" : deux pour la chaîne principale (squelette) et un pour la chaîne latérale.

Depuis ce modèle initial, de nombreux autres modèles ont été développés [155, 231, 242]. Par exemple, Voth et Izvekoc [117] ont développé un potentiel multi-échelle gros-grain où les paramètres du champ de force sont extraits de simulations de dynamique moléculaire en utilisant un procédé d'adaptation des forces. Souvent, les paramètres obtenus sont tabulés de façon à ne pas être restreints à la forme analytique d'un potentiel. Plus récemment, d'autres potentiels du même type ont été développés [105], permettant de simuler le repliement de protéines de petite taille [106].

Les modélisations gros-grain rendent les calculs plus rapides. Par exemple, pour

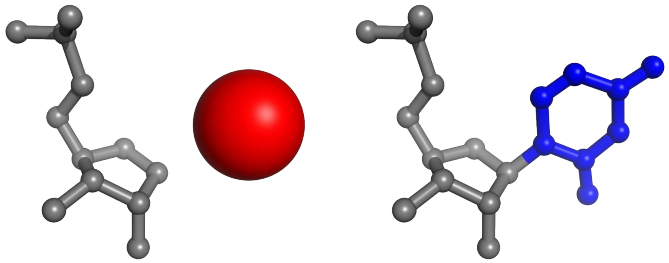
7. http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/

le passage d'un modèle tout atome à un modèle réduit avec un point par résidu/acide nucléique, le nombre d'atomes (pseudo-atomes) est environ divisé par dix, ce qui peux rendre les calculs jusqu'à 100 fois plus rapide. Le même ordre de grandeur peut être observé lors du passage d'un modèle réduit avec un point par résidu à un point par élément de structure secondaire.

Toutefois, le degré de précision nécessaire à la représentation fine des processus biologiques est souvent atomique. Il est donc difficile de dériver une représentation gros-grain générique pour un ensemble de systèmes et de simulations rendant la modélisation peu coûteuse en temps de calcul et précise quant à la représentation fine d'un phénomène biologique. Pour cette raison, de nombreux groupes ont combiné des approches à gros grains et grains fins dans des simulations mixtes [5]. Une revue des différentes approches est disponible [85].

Dans cette étude, je présenterai à nouveau rapidement le modèle utilisé par le logiciel RosettaDock puis, de façon plus détaillée, le modèle de Voronoï utilisé pour l'amarrage.

Dans le cadre de l'utilisation de RosettaDock, comme indiqué dans la section 2.1.1.3 du chapitre 2, la représentation gros-grain des protéines comprend l'ensemble des atomes du squelette et un à trois atomes pour la chaîne latérale appelés centroïdes [99]. Dans cette étude, nous nous sommes limités à l'utilisation du modèle à un centroïde que nous avons étendu à l'ARN et à l'ADN. Les coordonnées spatiales des atomes gros-grain des acides nucléiques sont calculées à partir des coordonnées atomiques des acides nucléiques. L'ARN et l'ADN sont donc pourvus des coordonnées spatiales de leurs atomes gros-grain (voir fig. 4.1).



a) Représentation géométrique gros-grain b) Représentation géométrique atomique

FIGURE 4.1 – Modèle gros-grain et modèle atomique : exemple de l'uracile. Le groupement phosphate et les atomes lourds du sucre sont représentés en gris. a) Le modèle gros grain avec le centroïde en rouge b) Le modèle atomique avec les atomes de la base en bleu. Le centroïde est le centre géométrique des atomes lourds.

Pour l'adénine, par exemple, nous passons d'un modèle avec 33 atomes à un modèle avec 13 atomes gros-grains. Le même facteur 3 est observé pour les autres acides nucléiques.

La taille des centroïdes des acides nucléiques ainsi remplacés est importante, car elle conditionne la distance à partir de laquelle on peut considérer une interpénétration des structures. Cette taille doit permettre aux centroïdes d'occuper l'espace des atomes

que chacun d'entre eux représente, pour leur permettre de se comporter de façon semblable. La taille des centroïdes est donc fixée à la valeur donnée pour leur version atomique, à savoir 8 Å.

Pour les protéines, dans le cadre du modèle de Voronoï, nous avons choisi de travailler avec un seul point par résidu, le centre géométrique de la chaîne latérale et du Cα. Ce choix permet tout d'abord de travailler sur un nombre réduit de points, mais aussi de s'affranchir des mouvements des chaînes latérales. En effet, même lorsque la chaîne latérale bouge, ce qui est fréquent à la surface de la protéine, le centre géométrique n'est souvent pas trop affecté. Pour la suite, la cellule de Voronoï correspondante est donc presque identique.

La triangulation de Delaunay pour les protéines a été le plus souvent effectuée à partir des carbones α [71, 183, 233, 253]. Nous avons choisi de construire les diagrammes de Voronoï à partir des centres géométriques des acides aminés, ceux-ci ayant permis d'obtenir de bons résultats dans le cadre de l'amarrage protéine-protéine [18, 15, 26, 27].

Ceux-ci ont été définis comme centres de gravité des atomes de la chaîne latérale et du carbone α, les atomes d'hydrogène n'étant pas pris en compte. Le centre géométrique ainsi obtenu est donc très proche du centre de masse de l'acide aminé.

Ce choix est à rapprocher d'une étude réalisée par S. Karlin débutée en 1994 [131, 130] dans laquelle trois types de distances sont étudiées :

- entre carbones α ;
- entre centres de gravité des acides aminés sans prendre en compte les carbones α ;
- entre centres de gravité de tous les atomes du résidu (chaîne latérale et squelette).

Ces trois types de distances sont utilisés pour mesurer la distance entre deux résidus dans un ensemble de structures tridimensionnelles de protéines connues. Cette étude statistique montre que les distances entre centres de gravité des chaînes latérales sont très sensibles aux interactions électrostatiques et hydrophobes, mais très peu aux contraintes stériques, contrairement aux distances entre les centres de gravité de tous les atomes de chaque résidu. Le fait de prendre en compte le carbone α dans le calcul du point qui va représenter chaque acide aminé, permet donc d'obtenir des propriétés intermédiaires. S. Karlin et ses collaborateurs montrent également que les distances entre carbones α sont largement décorrélées, à la fois des interactions et des contraintes stériques.

Le même modèle a été utilisé pour l'ARN où nous avons aussi un atome gros-grain par acide nucléique, situé au centre géométrique des atomes lourds de la base. En effet, le groupement phosphate et les atomes du sucre sont considérés comme faisant partie du squelette, tandis que les atomes lourds de la base sont considérés comme étant la chaîne latérale.

Contrairement aux protéines, qui n'ont que quatre atomes lourds dans leur squelette, les ARN ont douze atomes lourds et donc un squelette de taille importante. Ceci a pour conséquence de placer le centroïde très loin du squelette et donc de donner une grande importance à l'orientation de la base. Cette approche permet ainsi de représenter géométriquement la flexibilité de l'ARN pour mieux en tenir compte dans les mesures géométriques, plutôt que de placer un atome gros-grain au centre de

l'ensemble des atomes lourds de chaque nucléotide.

4.1.2 Mesure des termes géométriques à l'échelle gros-grain

Les termes géométriques sont uniquement utilisés ici pour les fonctions de score à l'échelle gros-grain. Seuls les acides aminés et les acides nucléiques à l'interface sont considérés pour calculer les termes géométriques. Ces termes ont déjà fait leurs preuves pour l'amarrage protéine-protéine [15]. Ils sont calculés à partir du diagramme de Voronoï.

Le diagramme de Voronoï est un pavage unique de l'espace en cellules de Voronoï à partir d'un ensemble de sites. Dans le cas qui nous intéresse, les sites sont les résidus ou pseudo-atomes. Le pavage est défini de telle sorte que n'importe quel point de l'espace plus proche d'un site que de tout autre appartient à la cellule de Voronoï de ce site.

Formellement, soit l'ensemble des n sites $\{p_j\}_{1\leq j\leq n}$, l'ensemble des positions spatiales $V(p_i)$ qui sont plus proches du site p_i que de tout autre site constitue la cellule de Voronoï du site p_i (voir éq. 4.1).

$$V(p_i) = \{x \in \mathbb{R}^d : \|x - p_i\| \leq \|x - p_j\|; \forall j \in \mathbb{N}, 1 \leq j \leq n\} \tag{4.1}$$

Les jointures entre les cellules de Voronoï s'appellent des facettes de Voronoï. En 3 dimensions, une cellule de Voronoï est un volume et une facette de Voronoï est une surface. Les facettes de Voronoï représentent la surface d'interaction entre les différents sites. Plus les sites sont proches et peu encombrés par les sites environnants, plus elles sont grandes. La cellule de Voronoï représente la zone où le site a le plus d'influence par rapport aux autres sites.

Intuitivement, la construction du diagramme de Voronoï utilise entre les différents sites le tracé des médiatrices. Pour plus de clarté, l'exemple illustré montre la construction manuelle d'un diagramme de Voronoï dans un espace en 2 dimensions (voir fig. 4.2).

La *Computational Geometry Algorithms Library* (CGAL [36]) permet la construction du diagramme de Voronoï. CGAL est une librairie implémentée en C++ de géométrie computationnelle algorithmiquement efficace. En pratique, pour une plus grande rapidité de calcul, CGAL obtient le diagramme de Voronoï en construisant son dual, la triangulation de Delaunay (voir fig. 4.3).

La triangulation de Delaunay est une triangulation uniquement constituée de triangles de Delaunay [167]. Un triangle est un triangle de Delaunay si son cercle circonscrit ne contient aucun autre site que les trois sommets du triangle. Dans une triangulation de Delaunay, tous les sites sont reliés par des segments de droites formant des triangles. L'algorithme construisant la triangulation de Delaunay en 3 dimensions est un algorithme incrémental probabiliste de complexité $\mathcal{O}(n^{\lceil \frac{3}{2} \rceil})$.

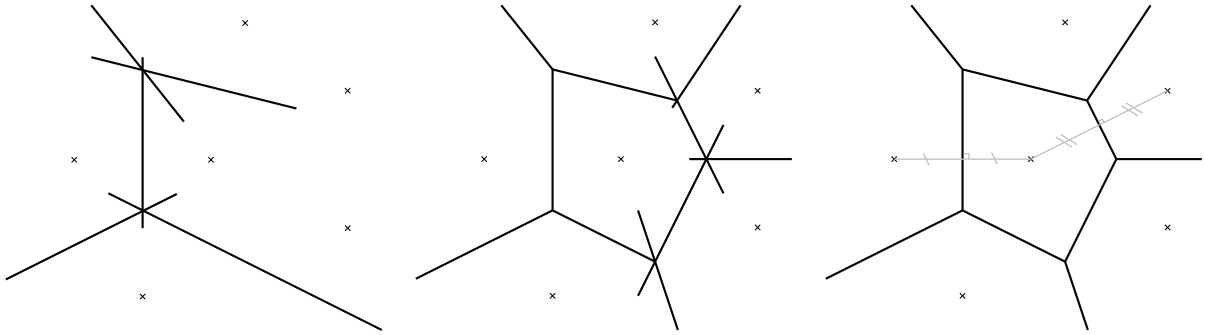


FIGURE 4.2 – Illustration de la construction intuitive d'un diagramme de Voronoï en 2 dimensions : a) tracer les médiatrices entre chaque paire de sites (représentés par des croix), b) puis limiter les segments de droite à leur jointure et c) itérer jusqu'à obtenir toutes les cellules de Voronoï.

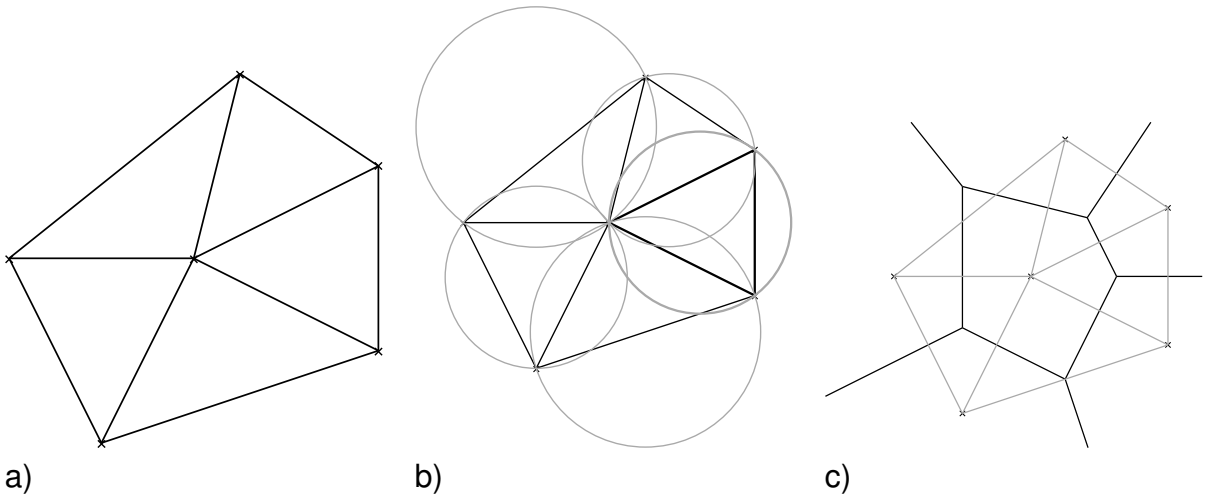


FIGURE 4.3 – Exemple de triangulation de Delaunay (en 2 dimensions), le dual du diagramme de Voronoï. a) Les droites sécantes entre les différents sites forment des triangles. b) Pour qu'une triangulation soit une triangulation de Delaunay, le cercle circonscrit de chaque triangle ne doit contenir aucun autre site. c) Pour un même ensemble de sites, en gris sont représentés les triangles de la triangulation de Delaunay et en noir les facettes de Voronoï.

La construction d'une triangulation de Delaunay commence par la sélection aléatoire de 3 sites, qui sont toujours une triangulation de Delaunay. Puis, on teste l'insertion aléatoire d'un nouveau site dans la triangulation en formant de nouveaux triangles avec ce site. Si le nouveau site inséré forme un triangle dont le cercle circonscrit contient un autre site, alors on teste la formation d'autres triangles avec le même site. Sinon, on accepte le nouveau site. On itère jusqu'à ce que tous les sites soient insérés

(voir fig. 4.4).

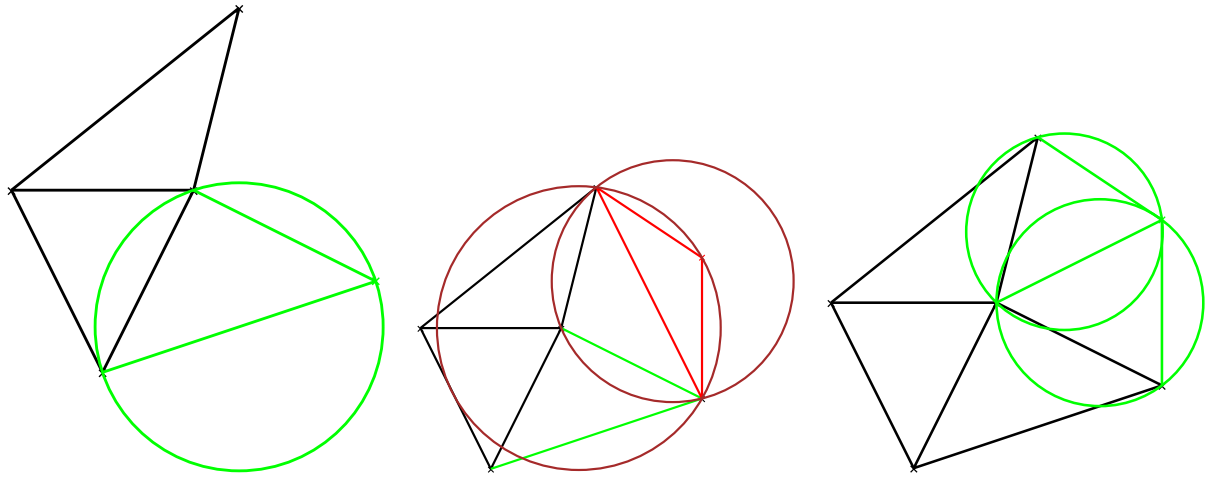


FIGURE 4.4 – Construction d'une triangulation de Delaunay. Cas initial : 3 sites traçant un triangle constituent toujours une triangulation de Delaunay. Étape d'insertion d'un nouveau site : a) insérer aléatoirement un nouveau site, et donc un nouveau triangle (l'insertion est valide si aucun autre site que les sommets du triangle n'est dans son cercle circonscrit) ; b) si un autre site appartient au cercle circonscrit, on refuse l'insertion du triangle et c) on teste alors la construction d'un autre triangle.

CGAL est couplé à la *Easy Structural Biology Templated Library* (ESBTL [168]) pour déterminer l'interface et obtenir les mesures statistiques. ESBTL est une librairie implémentée en C++, munie de *templates* pour modéliser et manipuler efficacement les entités biologiques. Il est notamment possible de sélectionner des entités biologiques telles que des chaînes d'acides aminés ou d'acides nucléiques, des acides aminés, des acides nucléiques, des atomes. Des filtres peuvent être utilisés pour ne sélectionner qu'un sous-ensemble d'entités biologiques en fonction de son type. Nous avons étendu ESBTL pour :

- définir des acides nucléiques ;
- construire automatiquement des atomes gros-grain à partir d'atomes de l'échelle atomique ;
- calculer l'aire d'une facette de Voronoï et le volume d'une cellule de Voronoï en 3D.

L'interface peut être définie grâce au diagramme de Voronoï. L'usage d'une structure géométrique telle que le diagramme de Voronoï permet en effet de s'affranchir des seuils arbitraires de distance utilisés pour définir si deux atomes sont en interaction. Ainsi, au sens du diagramme de Voronoï, pour que deux atomes gros-grain soient en interaction, ils doivent partager une facette de Voronoï. L'interface est de cette manière représentée par l'ensemble des atomes gros-grain partageant une facette de

Voronoï avec au moins un atome gros-grain de l'autre partenaire (voir fig. 4.5). C'est cette définition de l'interface qui est utilisée pour la fonction de score géométrique.

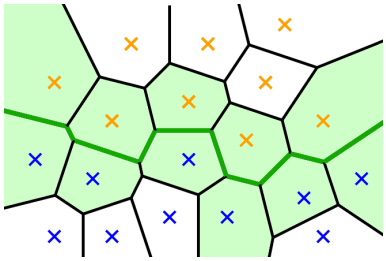


FIGURE 4.5 – L'interface d'une tessellation de Voronoï appliquée à une interaction protéine-ARN. Les atomes gros-grain de la protéine (en bleu) et de l'ARN (en orange) forment à leur interaction une interface (en vert).

Le solvant est ajouté explicitement aux structures 3D notamment à l'aide de ESBTL. Le solvant explicite a pour fonction de délimiter les surfaces extérieures des cellules de Voronoï qui ne sont pas enfouies dans la structure. L'ajout du solvant se traduit par l'insertion de sites supplémentaires, avant la construction du diagramme de Voronoï. Ces sites supplémentaires sont insérés sous la forme d'un treillis où tout site est distant de tous ses voisins de 5 Å. Comme l'ajout du solvant explicite a pour conséquence une forte augmentation du temps de calcul du diagramme de Voronoï, les atomes gros-grain de solvant inutiles sont retirés. Sont inutiles tous les atomes gros-grain de solvant qui sont trop loin des atomes gros-grain de chacun des deux partenaires pour interagir avec eux. Comme l'ajout du solvant explicite peut induire un biais dans les résultats, les atomes gros-grain en interaction avec le solvant ne sont pas considérés comme faisant partie de l'interface.

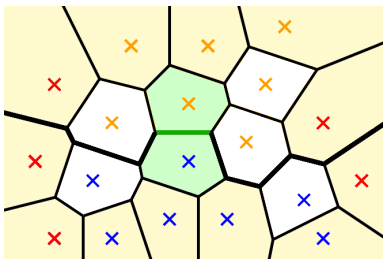


FIGURE 4.6 – Le solvant à l'interface d'une tessellation de Voronoï construite sur les atomes gros-grain d'une interaction protéine-ARN. Les atomes gros-grain du solvant explicite (en rouge) réduisent l'interface (en vert) : tout atome gros-grain interagissant avec un atome gros-grain du solvant est considéré en dehors de l'interface.

4.1.3 Termes géométriques à l'échelle gros-grain

Il y a 210 termes de score géométriques répartis en 6 types de termes de score géométriques (voir fig. 4.7) :

- P1 (1 terme), le nombre de centroïdes à l'interface ;
- P2 (1 terme), l'aire totale d'interaction, définie par la somme des aires des facettes de Voronoï à l'interface ;
- P3 (24 termes), la proportion des cellules de Voronoï des centroïdes à l'interface sur le nombre total de cellules de Voronoï à l'interface, pour chaque type d'acide aminé (20) ou d'acide nucléique (4) ;
- P4 (24 termes), la volume médian des centroïdes à l'interface, pour chaque type d'acide aminé (20) ou d'acide nucléique (4) ;
- P5 (80 termes), la proportion des facettes de Voronoï à l'interface sur le nombre total de facettes de Voronoï à l'interface, pour chaque paire de types d'acides aminés (20) et d'acides nucléiques (4) ;
- P6 (80 termes), la distance médiane entre les centroïdes à l'interface de chacun des deux partenaires, pour chaque paire de types d'acides aminés (20) et d'acides nucléiques (4).

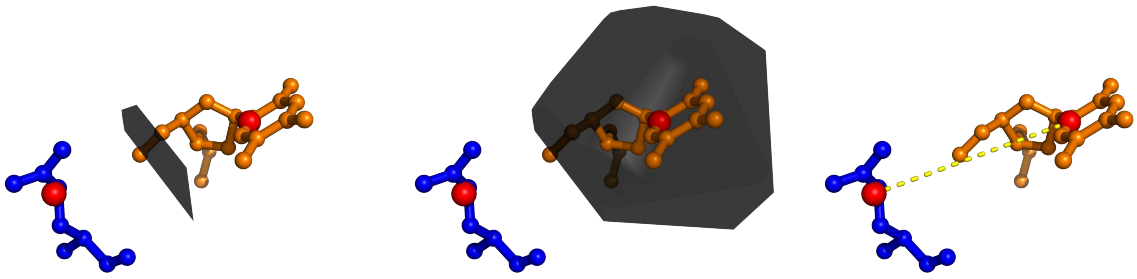


FIGURE 4.7 – Mesure des paramètres gros-grain à partir du diagramme de Voronoï entre une protéine (en bleu) et un ARN (en orange). a) Facette de Voronoï (surface grise) entre un acide aminé et un acide nucléique en interaction. Le paramètre P2 correspond à la somme des aires des facettes de Voronoï constituant l'interface. b) Cellule de Voronoï (en gris et ici découpée pour faire apparaître l'ARN en son centre) d'un acide aminé. Le paramètre P4 d'un type d'atome gros-grain correpond à la valeur médiane de ce volume sur tous les atomes gros-grain du type donné. c) Distance entre deux atomes gros-grain (en jaune). Le paramètre P6 de deux types donnés d'atomes gros-grain correspond à la valeur médiane de cette distance sur toutes les paires d'atomes gros-grain de ces deux types donnés.

On peut déjà remarquer que le nombre de termes de score est élevé, avec 210 termes. D'autres paramètres auraient pu être utilisés, comme le nombre de facettes de Voronoï ou la distance médiane entre tous les centroïdes à l'interface. Mais ces

termes de score auraient pu augmenter la redondance des informations contenues dans les différents termes de score.

Les deux premiers termes de score P1 et P2 sont attendus plus grands pour les presque-natifs que pour les leurres. Cependant, il est tout à fait possible que leurs valeurs oscillent autour d'une valeur moyenne pour les presque-natifs et que des candidats puissent avoir des valeurs encore plus élevées sans pour autant correspondre à des presque-natifs. Le terme de score P1 correspond à une version issue du diagramme de Voronoï du terme de score gros-grain interchain_contact.

Les termes de score P3 et P5 ont pour objectif de mesurer l'impact de la présence de chacun des types d'acide aminé ou d'acide nucléique à l'interface (P3), mais aussi leurs interactions préférentielles (P5). En effet, si un terme de score P3 est sensiblement plus élevé au sein des presque-natifs qu'au sein des leurres, cela peut signifier que le type d'acide aminé ou d'acide nucléique auquel il correspond se trouve préférentiellement à l'interface. De la même manière pour un terme de score P5, cela signifierait que le type d'acide aminé et le type d'acide nucléique qui lui correspondent sont plus souvent que les autres en interaction dans les complexes protéine-ARN. Les termes de score P3 et P5 correspondent respectivement à une version géométrique des termes de score interchain_env et interchain_pair, mais dédiés à l'interface.

Le terme de score P4 évalue spécifiquement l'empilement stérique. Les acides aminés et nucléiques au contact les uns des autres forment ce qui est appelé un empilement stérique. Si les atomes gros-grain sont trop proches les uns des autres, le volume médian sera petit. Or, certains acides nucléiques et acides aminés occupent plus d'espace que d'autres. On attend donc des atomes gros-grain des acides aminés et acides nucléiques dont la chaîne latérale est petite d'avoir un petit volume médian chez les presque-natifs, et inversement.

Le terme de score P6 a plusieurs usages. D'une part, la distance médiane permet de repérer les structures pour lesquelles une interaction anormalement longue intervient, présageant d'un cas pathologique de leurre (mauvais empilement et donc possiblement un problème lors de la génération des leurres). D'autre part, P6 permet de retrouver l'orientation des acides aminés et acides nucléiques et d'évaluer si l'orientation de l'interaction se fait telle qu'attendue dans un presque-natif. En effet, les atomes gros-grain représentent la chaîne latérale et, selon l'orientation des acides aminés et acides nucléiques à l'interaction, la distance entre eux n'est pas la même.

4.1.4 Données et fonctions de score à l'échelle gros-grain

Le jeu de données atomique de candidats est généré autour et à proximité de la solution. Or, la fonction de score gros-grain a besoin de discriminer des candidats même très éloignés de la solution des candidats modélisant l'épitope de l'interaction. Ce jeu de données de candidats générés n'est donc pas utilisable pour l'optimisation de la fonction de score gros-grain.

Un jeu de données de candidats est donc généré à l'identique du jeu de données atomique, aux paramètres de la perturbation près. Plutôt que de définir un nuage de perturbation gaussienne, les coordonnées de l'ARN sont définies aléatoirement au-

tour de la protéine. Ceci assure que la structure native n'affecte pas le résultat et que n'importe quel candidat puisse être généré et utilisé pour l'apprentissage comme pour l'évaluation.

Les candidats générés sont ensuite étiquetés par un seuil différent en IRMSD de celui utilisé pour le jeu de données atomique, pour refléter la différence d'objectif. L'objectif de la fonction de score gros-grain est en effet de trouver l'épitope, pas d'opérer un raffinement de la structure. Le seuil de IRMSD choisi est de 12 Å, seuil en-dessous duquel un candidat est appelé *presque-natif*, par analogie avec l'échelle atomique.

4.2 Optimisation de la fonction de score gros-grain

Comme pour l'échelle atomique, une fonction de score gros-grain implémentée directement dans RosettaDock nécessite d'être modélisée sous la forme d'une combinaison linéaire des paramètres. Il est cependant possible de s'affranchir d'une telle contrainte et de n'utiliser la fonction de score gros-grain qu'en post-traitement de la génération. Une telle utilisation implique d'avoir déjà généré les candidats. Deux approches sont donc envisageables : l'utilisation d'une combinaison linéaire ou la mise en place d'une fonction de score *a posteriori*.

ROGER, étant données ses performances pour l'adaptation de la fonction de score atomique de RosettaDock à la prédiction d'interactions protéine-ARN, est utilisé pour l'apprentissage d'une fonction logistique pour modéliser une fonction de score gros-grain. La fonction de score ainsi apprise est appelée *VOR*. Nous allons maintenant détailler comment cette fonction de score est apprise au moyen de ROGER.

4.2.1 Méthodologie d'optimisation de la fonction de score gros-grain

La méthodologie générale d'optimisation de la fonction de score gros-grain se décline en cinq phases :

- la génération de candidats issus des complexes de la PRIDB par perturbation, mais avec des candidats divergeant davantage de la structure native ;
- la construction d'un diagramme de Voronoï sur chacun des candidats générés ;
- la mesure des valeurs des termes de score géométriques gros-grain sur les diagrammes de Voronoï construits ;
- l'apprentissage de la fonction de score sur l'ensemble des 120 complexes de la PRIDB ;
- l'évaluation des 120 fonctions de score apprises en *leave-"one-pdb"-out* et évaluées sur les 10 000 candidats de leurs jeux d'évaluation.

La génération de candidats pour l'amarrage gros-grain doit donner des candidats divergeant davantage de la solution. De plus, cette génération ne doit pas tenir compte de la structure du natif. C'est pourquoi la génération des candidats de l'amarrage gros-grain se fait en aveugle : les coordonnées spatiales de l'ARN sont déterminées aléatoirement. Une telle génération sur 10 000 candidats nous empêche

par conséquent d'imposer la même contrainte du nombre minimum de presque-natifs et de leurres générés par complexe. Si le nombre de leurres est rapidement important, le nombre de presque-natifs générés reste très faible. Cependant, nous voulons que la fonction de score gros-grain nous permette seulement de prédire les structures suffisamment proches de la solution pour être affinées par la fonction de score atomique. Nous pouvons donc considérer à l'échelle gros-grain et pour des raisons d'apprentissage que le seuil de presque-natifs est cette fois-ci de 10 Å. Nous pouvons vérifier que des candidats à moins de 10 Å de la structure native en IRMSD permettent bien d'affiner la structure par protocole d'amarrage grâce aux entonnoirs détectés pour la fonction de score atomique (voir fig. 4.8).

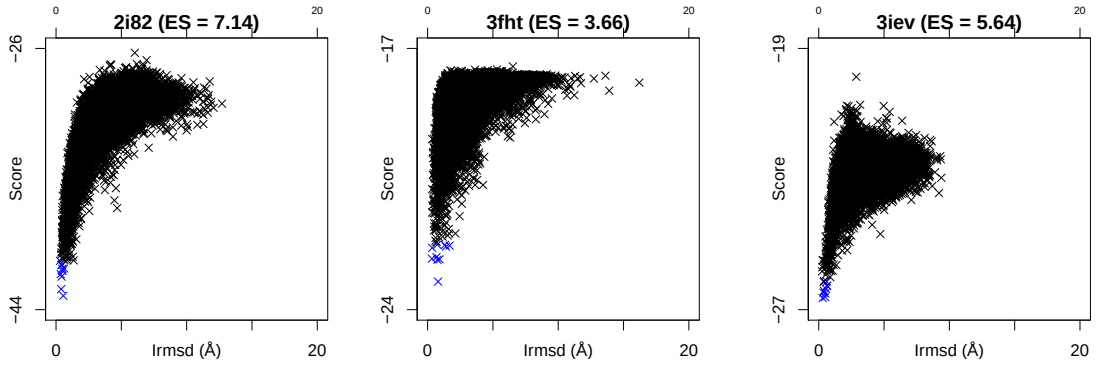


FIGURE 4.8 – Exemples de diagrammes d'énergie en fonction du IRMSD pour la fonction de score POS sur 3 complexes. On remarque des entonnoirs permettant un affinement de structures candidates trouvées à moins de 10 Å de la structure native.

Contrairement aux termes de score physico-chimiques, ici, tous les termes de score géométriques ont des valeurs positives ou nulles. Nous ne contraignons donc pas l'apprentissage à rechercher les poids sur l'intervalle de valeurs positif avec cette fonction de score gros-grain. Nous étudions toutefois les différences entre un apprentissage à poids positifs et un apprentissage sans contrainte sur l'intervalle de définition des poids.

4.3 Évaluation de l'interaction à l'échelle gros-grain

La fonction de score gros-grain est modélisée au moyen de termes géométriques issus de la construction d'un diagramme de Voronoï avec pour sites de construction du diagramme les atomes gros-grain de la structure 3D. En plus d'établir un protocole permettant la prédiction des interactions protéine-ARN, il s'agit de mieux comprendre les mécanismes donnant lieu à une interaction entre protéine et ARN. C'est pourquoi nous étudions non seulement la fonction de score modélisée, mais aussi les valeurs des paramètres selon le type de structure rencontrée : native ou candidate.

4.3.1 Étude des valeurs des mesures géométriques

Les valeurs des mesures géométriques des structures natives et des candidats permettent d'en apprendre plus sur le comportement des mesures géométriques lorsqu'il y a interaction entre protéine et ARN. Toutes les valeurs des mesures géométriques sont ici analysées avant traitement des valeurs non renseignées : ces dernières ne sont tout simplement pas prises en compte pour cette analyse. Les mesures les plus éloquentes sont certainement le nombre de cellules de Voronoï en interaction et l'aire de la surface totale d'interaction pour une même structure 3D. Pour ces deux mesures, les valeurs sont réparties très différemment entre les structures natives et les candidats.

En effet, les structures natives ont une surface d'interaction minimale de 10 Å^2, contre 0 pour des candidats où les partenaires sont suffisamment éloignés l'un de l'autre (voir tableau 4.1). Rappelons qu'il est possible pour des candidats d'avoir une surface d'interaction nulle car tout atome gros-grain en interaction avec le solvant est retiré de l'interface. La médiane de la surface d'interaction des structures natives est de 756 Å^2, contre 336 Å^2 pour les candidats. La méthode de génération des candidats n'impose pas de contrainte sur les partenaires. Comme ceux-ci sont rigides, les perturbations engendrent de nombreux candidats qui présentent un mauvais emboitement et une interface faible (voir fig. 4.9).

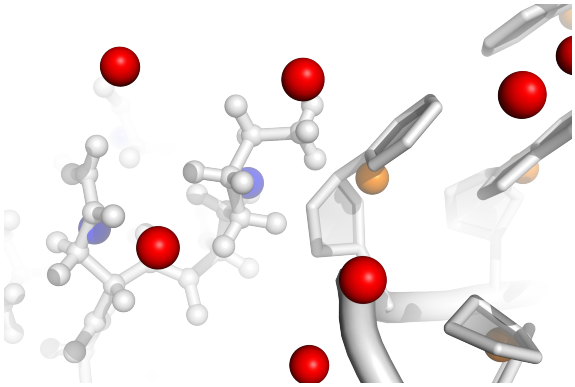


FIGURE 4.9 – Interaction de la protéine (en bleu) et de l'ARN (en orange) avec le solvant explicite (en rouge) du modèle gros-grain, avec le modèle atomique des deux partenaires affiché pour information (en gris). Il peut arriver comme ici que le nombre d'acides aminés et de résidus à l'interface soit réduit à 0 par la présence d'atomes gros-grain de solvant autour d'une petite interface composée de trois ou quatre acides aminés et nucléiques.

Pour le nombre de cellules de Voronoï à l'interface, c'est d'autant plus flagrant, puisque la médiane du nombre de cellules de Voronoï à l'interface est de 44 pour les structures natives, contre 28 pour les candidats. Il y a au minimum 3 cellules de Voronoï à l'interface dans les structures natives, contre 27 220 candidats qui n'en ont aucune. Ces 27 220 candidats sont répartis sur 26 complexes et ont des partenaires trop éloignés pour avoir une interface, le solvant se logeant trop près des quelques

atomes gros-grain interagissant. Cette absence d'interface est parfois aggravée par le traitement des ions. Les ions à l'interface sont en effet retirés de la structure 3D et sont par conséquent remplacés par des atomes de solvant s'ils laissent un vide trop important. Or, tout atome gros-grain en interaction avec un atome gros-grain du solvant n'est pas considéré comme faisant partie de l'interface, pour éviter que le solvant ne biaise les résultats des surfaces d'interaction et des volumes des cellules de Voronoï.

Agrégat	Aire de l'interface	Nombre de cellules
Maximum	3209	174
Moyenne	952	54
Médiane	756	44
Minimum	10	3

TABLE 4.1 – Répartition de deux termes de score géométriques sur les structures natives de la PRIDB. L'aire de l'interface (P1) est la surface totale de l'interaction en Å^2, qui est indiquée avec le nombre de cellules de Voronoï à l'interface.

Les autres types de mesures sont répartis en fonction du type d'acide aminé ou d'acide nucléique et sont donc plus nombreux, mais aussi parfois avec plus de valeurs manquantes. On peut toutefois remarquer certaines propriétés intéressantes en comparaison avec les interactions protéine-protéine. Il a été vu pour les interactions protéine-protéine que les acides aminés hydrophobes étaient les plus présents dans l'interface entre les partenaires [15]. Pour les interactions protéine-ARN, c'est le phénomène inverse qui est observé (voir fig. 4.11). À part la leucine, ce sont les acides aminés polaires – et notamment chargés – qui sont à l'interface : parmi les acides aminés hydrophobes, seule la leucine fait partie des acides aminés de proportion non nulle à l'interface sur plus de la moitié des structures natives. La leucine se partage plus de la moitié des interfaces des structures natives avec l'arginine, l'aspartate, le glutamate, la glycine, la lysine et la sérine. Au total, parmi les acides aminés chargés, seule l'histidine est absente de l'interface sur l'ensemble des structures natives alors que, au contraire, parmi les acides aminés polaires non chargés, ce sont la glycine et la sérine qui ont une proportion non nulle. Aucun acide aminé n'est à l'interface dans toutes les structures natives. De plus, seule la lysine est présente à l'interface dans plus des trois quarts des structures natives. De façon plus générale, on constate une queue de distribution plus élevée pour les leurres que pour les presque-natifs et natifs.

De la même manière, les uraciles représentent en moyenne 12 % des acides aminés et nucléiques à l'interface, bien au-delà de la guanine (8 %), la cystéine (7 %) et l'adénine (5 %). Et seul l'uracile est présent à l'interface dans plus des trois-quarts des structures natives (voir fig. 4.12). On peut donc en conclure que les acides aminés dominent largement les acides nucléiques dans leur participation à l'interaction. Il y a plus d'un acide aminé pour un acide nucléique. Le rapport est plus proche de sept acides aminés pour trois acides nucléiques dans les structures natives, donc plus de deux pour un. C'est un rapport important à garder en tête lorsque l'on souhaite modéliser une interaction protéine-ARN. En effet, cela signifie que ce sont les atomes

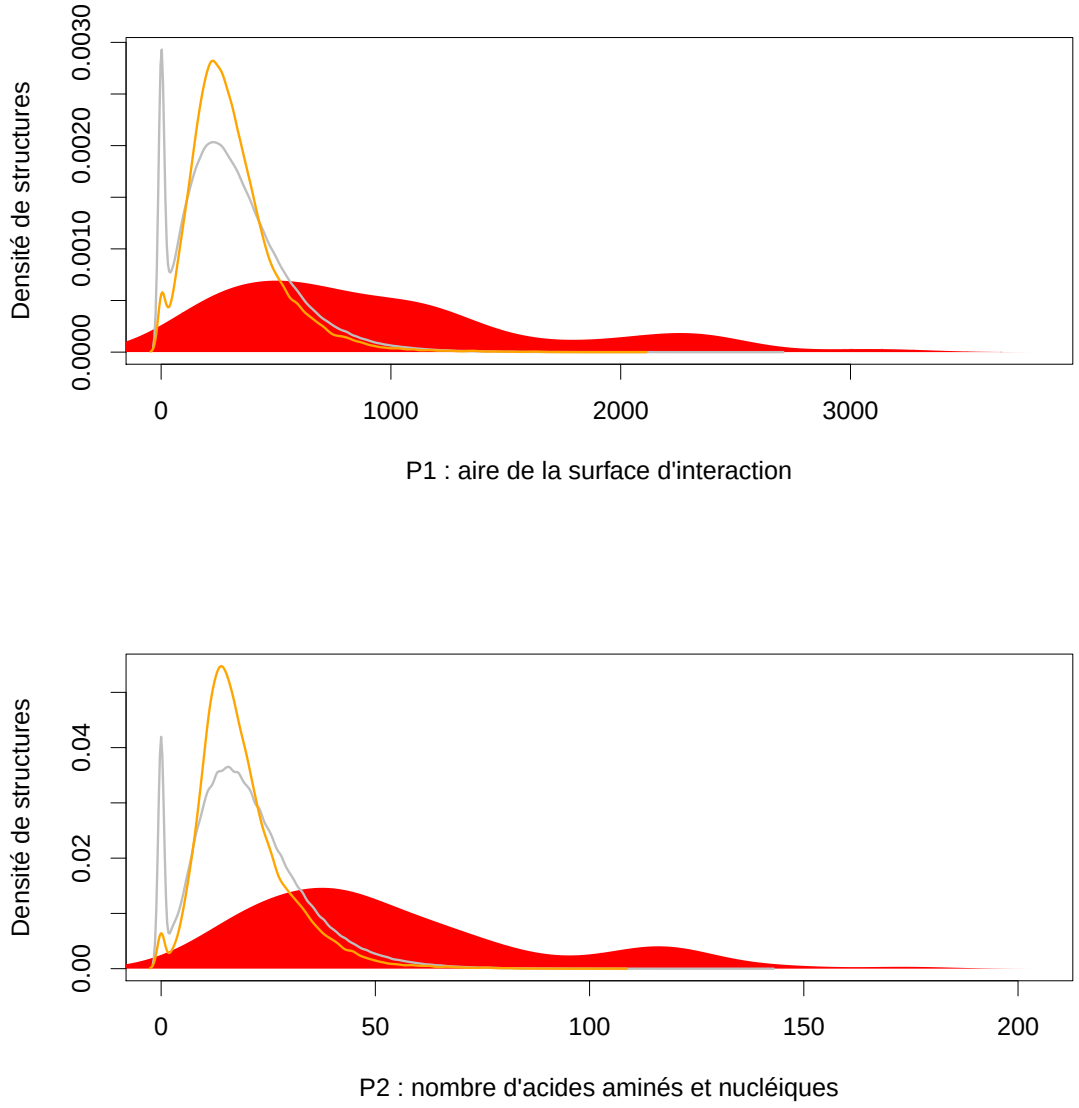


FIGURE 4.10 – Courbes de densité des termes de score P1 (surface d'interaction) et P2 (nombre d'acides aminés et nucléiques à l'interface) sur les 120 structures natives (en rouge), sur les 64 994 presque-natifs (en orange) et sur les 1 135 006 leurres (en gris). On peut remarquer que de nombreux candidats ne présentent pas d'interface, ce qui est dû au fait que tous les atomes gros-grain en interaction entre les deux partenaires sont aussi en interaction avec le solvant.

gros-grain des acides aminés qui ont le plus d'impact sur la valeur du score.

La forte proportion de candidats sans un ou plusieurs types d'acides nucléiques à l'interface est expliqué par deux phénomènes :

– les candidats sans interface ;

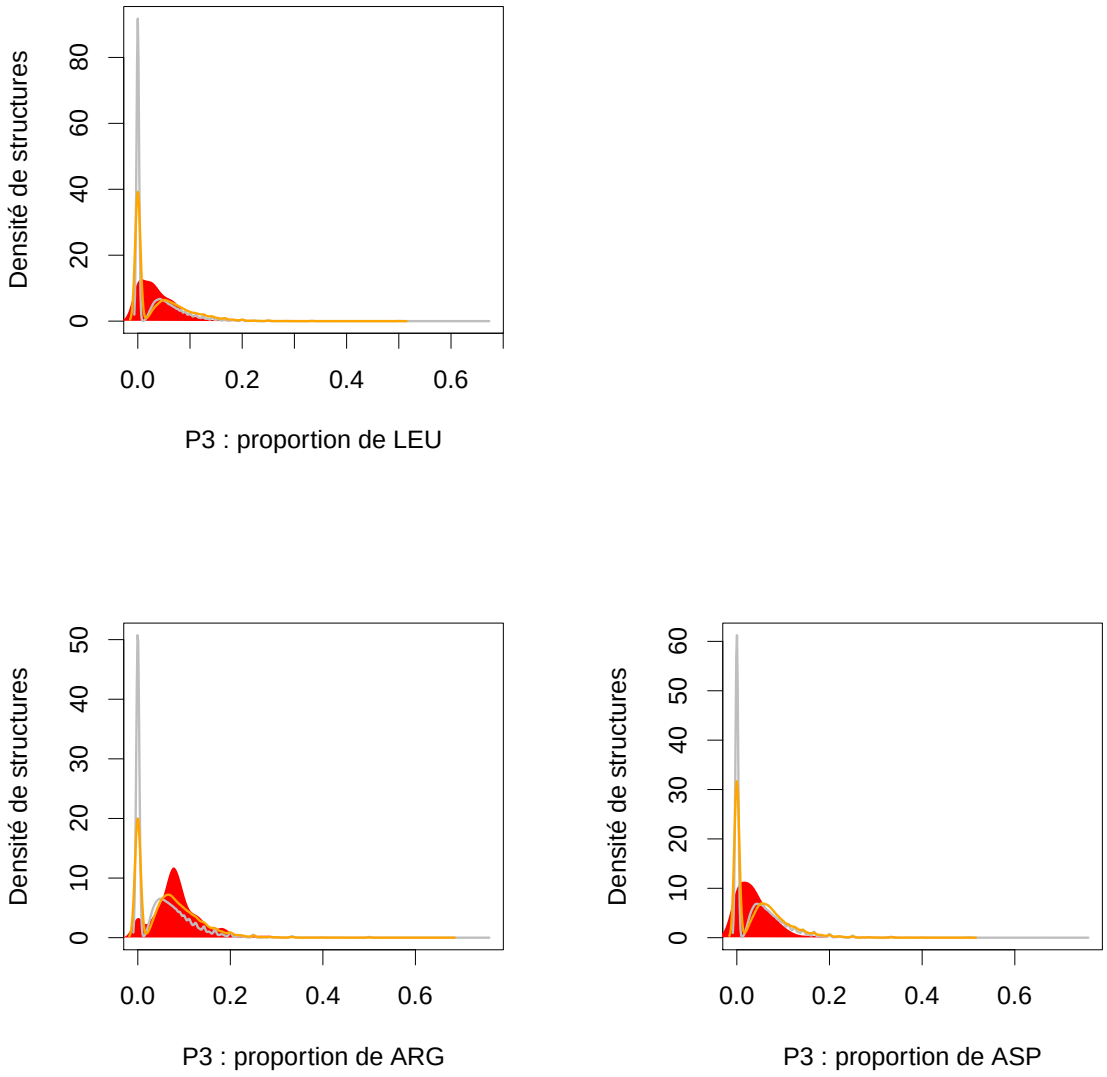


FIGURE 4.11 – Courbes de densité des termes de score P3 pour quelques acides aminés sur les 120 structures natives (en rouge), sur les presque-natifs (en orange) et sur les leurres (en gris).

– les candidats avec de petits ARN ou des ARN ne présentant qu'un ou deux types d'acides nucléiques.

Nous avons en effet vu que les candidats sans interface étaient nombreux et ont nécessairement les valeurs des quatre paramètres P3 des acides nucléiques à zéro. Mais il existe aussi de petits ARN ou des ARN ne présentant pas un ou plusieurs types d'acides nucléiques. C'est le cas des queues poly-A ou poly-U, qui ne possèdent que des adénines ou des uraciles. Il existe aussi des complexes dont l'ARN ne contient

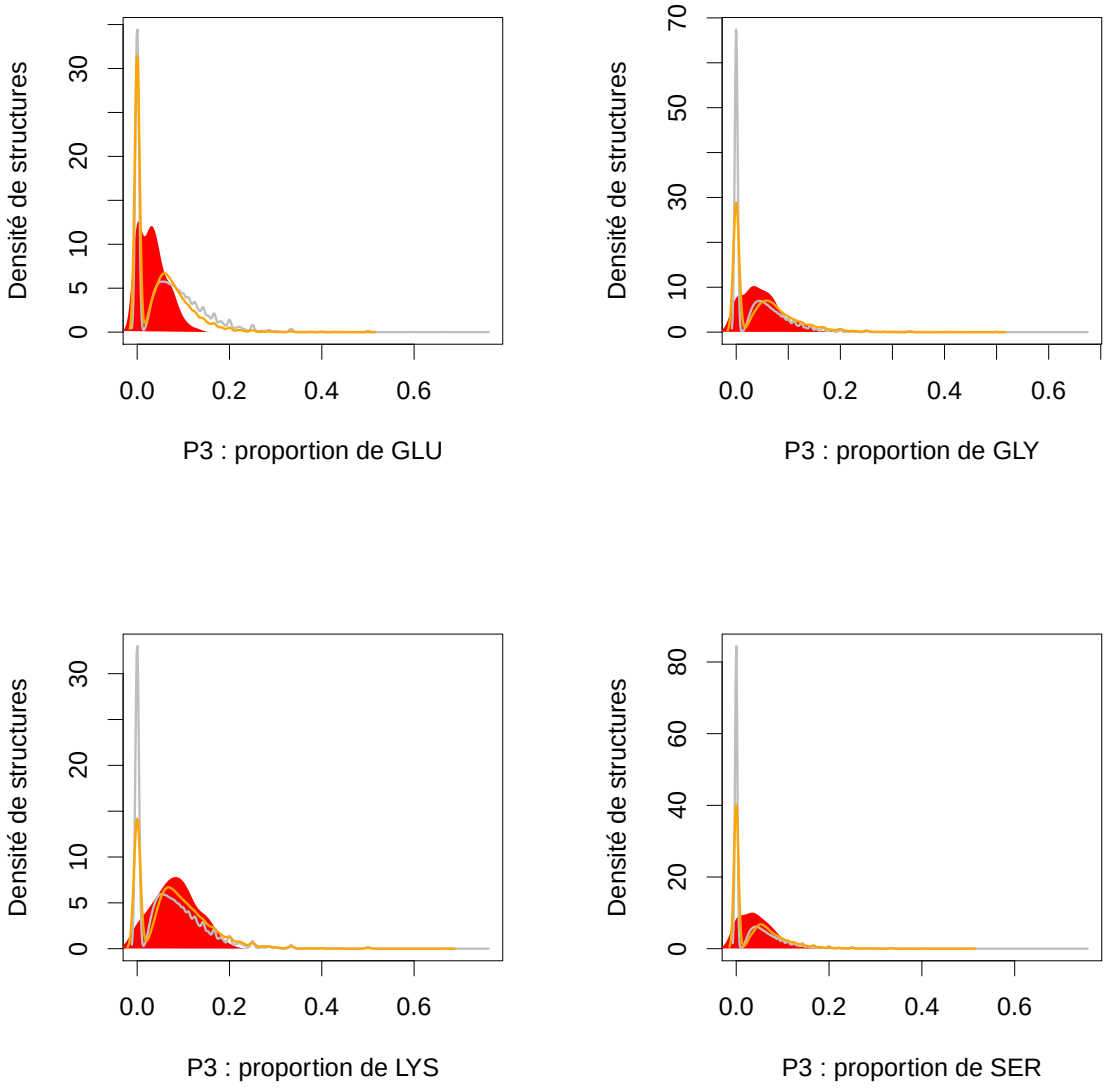


FIGURE 4.11 (cont.) – Courbes de densité des termes de score P3 pour quelques acides aminés sur les 120 structures natives (en rouge), sur les presque-natifs (en orange) et sur les leurres (en gris).

qu'une répétition du motif GC. Aucun de ces complexes ne possède certains types d'acides nucléiques et ceci se retrouve dans les paramètres P3 des acides nucléiques sous la forme d'un pic à l'abscisse zéro (voir fig. 4.12).

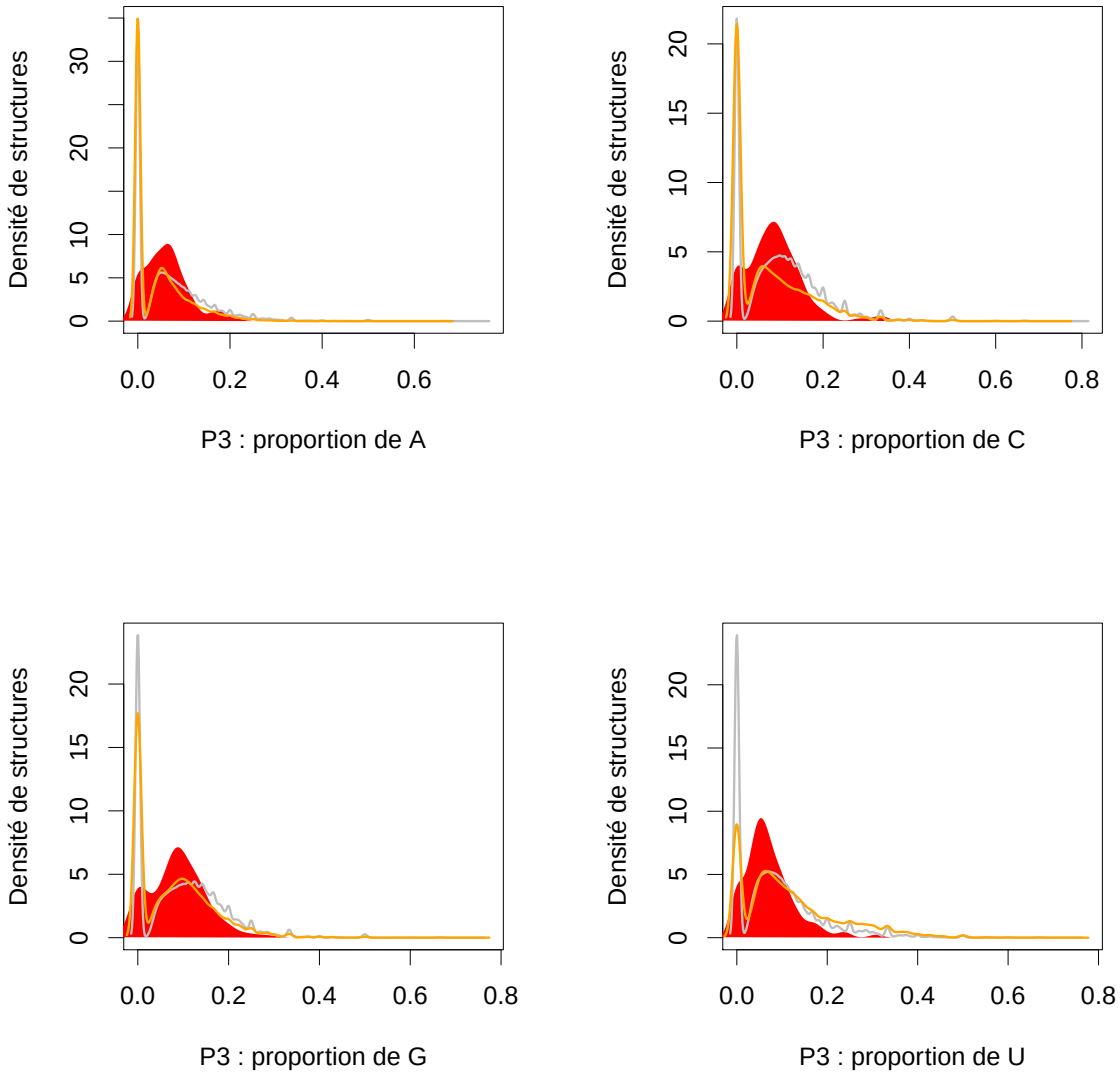


FIGURE 4.12 – Courbes de densité des termes de score P3 pour les acides nucléiques sur les 120 structures natives (en rouge), sur les presque-natifs (en orange) et sur les leurres (en gris). Ici, s'ajoutent aux candidats sans interface les candidats avec de très petits ARN, comme des queues poly-A, qui ne contiennent que des adénines et, par conséquent, ne possèdent pas d'autre acide nucléique à l'interface que des adénines.

4.3.2 Évaluation de la fonction de score gros-grain

La fonction de score gros-grain a pour objectif de trier les candidats de telle sorte qu'au moins une structure modélisant l'épitope de l'interaction fasse partie des premières structures du tri. Une fois l'épitope détecté, c'est à la fonction de score atomique

d'affiner la structure et d'en modéliser l'interaction 3D. Nous allons d'abord évaluer les poids de la fonction de score gros-grain, puis ses performances, que nous comparerons ensuite à la même fonction de score dont les poids sont appris dans l'intervalle de valeurs positif, ainsi qu'à une fonction de score non linéaire utilisant une valeur de centrage par terme de score (voir section 3.1.1 chapitre 3).

4.3.2.1 Poids

Les poids des paramètres P1 (l'aire totale de la surface d'interaction) et P2 (le nombre d'acides aminés et nucléiques à l'interface) sont sans surprise très négatifs (voir 4.13). Ces poids étaient attendus négatifs car ils sont censés favoriser l'interaction : il faut qu'il y ait un nombre suffisant d'acides aminés et nucléiques (pour P1), de même qu'une surface d'interaction suffisante (pour P2) pour obtenir une interaction entre les deux partenaires. On constate tout de même une distance d'interquartile (entre le premier quartile et le troisième) plus importante pour P2 que pour P1. Autrement dit, l'apprentissage sur chacun des plis du *leave-"one-pdb"-out* ne donne pas un consensus aussi clair pour l'utilité de P2 que pour celle de P1 pour prédire l'interaction protéine-ARN. Ceci suggère que le nombre d'acides aminés et nucléiques en interaction a une contribution plus discutable que la surface d'interaction.

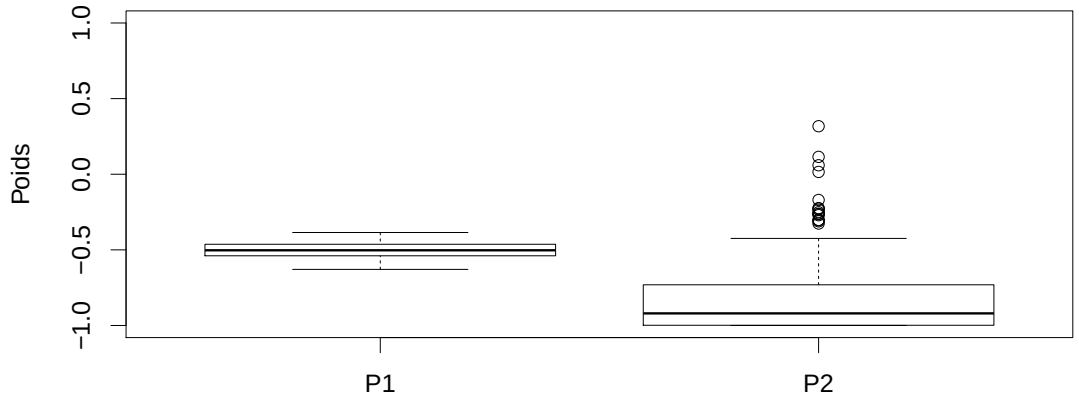


FIGURE 4.13 – Valeurs des coefficients des paramètres P1 et P2 pour les 120 complexes : l'aire totale de la surface d'interaction (en Å^2) et le nombre d'acide aminés et nucléiques à l'interface. P1 et P2 sont très négatifs.

On peut aussi constater que tous les paramètres P3 (proportion d'acides aminés à l'interaction) fluctuent autour d'une valeur de 0.5, avec peu de différences entre les types d'acides aminés ou nucléiques (voir fig. 4.14). Ces poids ont donc tendance à

augmenter le score final, donc classer le candidat comme leurre. La fonction de score a pris en compte le fait qu'il existe des leurres pour lesquels la proportion d'acides aminés de même type à l'interaction est très importante. Cette proportion importante d'acides aminés de même type correspond à la queue de distribution de chacun des termes de score de P3 vue à la section précédente.

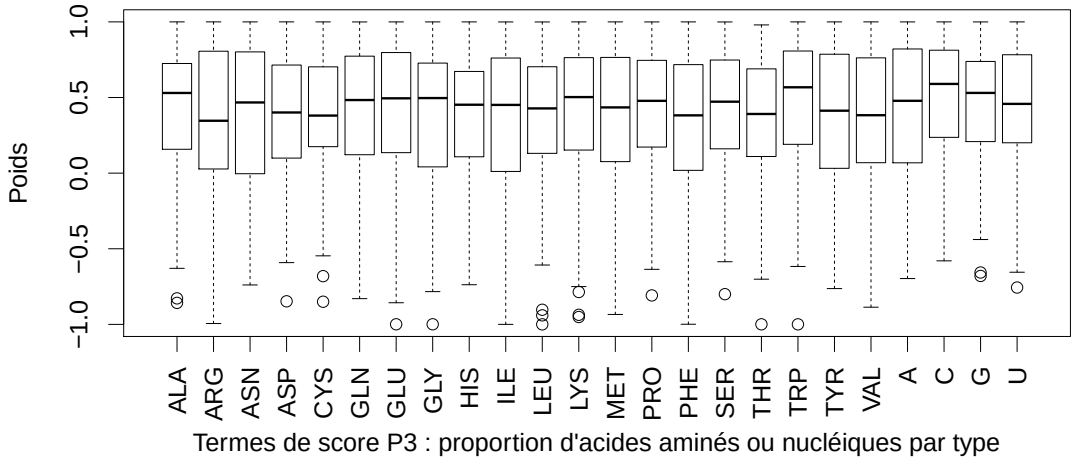


FIGURE 4.14 – Valeurs des coefficients des paramètres P3 pour les 120 complexes : les proportions de chaque type d'acide aminé et d'acide nucléique. P3 pénalise les structures qui privilégient à leur interface la présence d'un type d'acide aminé ou nucléique plutôt qu'un autre.

Comparativement aux poids appris pour les interactions protéine-protéine, les poids des paramètres P4 – les volumes médians – ne sont pas tous négatifs (voir 4.15). Au contraire, plusieurs acides aminés montrent des poids positifs de grande amplitude. En réalité, les poids P4 montrent que l'ordonnancement des acides aminés selon leur taille (ou encombrement stérique) a partiellement été retrouvé à travers l'apprentissage. Ils sont positifs pour les acides aminés de petite taille (alanine, glycine, aspartate, glutamate, méthionine, valine) et négatifs de grande amplitude pour les acides aminés de taille importante (arginine, tyrosine, tryptophane, phénylalanine, histidine). Les acides aminés polaires non chargés (asparagine, glutamine) de composition proche de certains résidus chargés ont aussi des poids négatifs de plus grande amplitude que les acides aminés chargés correspondants : l'aspartate et le glutamate déjà cités plus haut. Globalement, les poids des paramètres P4 sont négatifs ou nuls même pour des acides aminés de taille moyenne (thréonine, leucine, isoleucine, lysine, sérine, cystéine, proline). Pour les acides nucléiques, les poids sont positifs et faibles, indiquant que la fonction de score favorise les candidats ayant un volume médian relativement faible pour chaque acide nucléique, en comparaison des volumes médians

obtenus sur les candidats d'une génération de candidats par perturbation. Les poids des pyrimidines (G et C) sont plus élevés que ceux des purines, qui sont de taille plus petite relativement aux pyrimidines. Le même constat que pour les protéines peut donc être fait pour les acides nucléiques : l'ordonnancement des acides nucléiques selon leur taille a été appris par la fonction de score. D'un point de vue géométrique, les acides aminés et nucléiques avec une taille importante ont tendance à obtenir des cellules de Voronoï plus volumineuses dans une structure native ou presque-native. Dans une structure générée aléatoirement et éloignée de la solution native, la cellule de Voronoï peut avoir un volume plus élevé ou plus faible. Cette différence entre le volume des cellules de Voronoï pour les presque-natifs et pour les leurres lors de l'apprentissage a donc permis à l'algorithme d'apprentissage de retrouver les tailles des différents acides aminés et nucléiques.

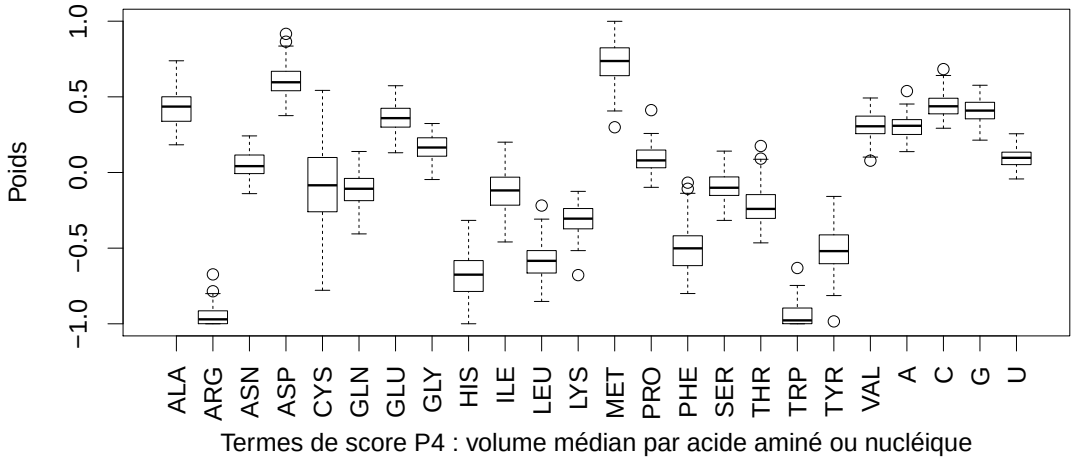


FIGURE 4.15 – Valeurs des coefficients des paramètres P4 pour les 120 complexes : les volumes médians par type d'acide aminé ou nucléique. Les poids montrent que la fonction de score a retrouvé l'information d'encombrement stérique de chaque type d'acide aminé et nucléique.

4.3.2.2 Évaluation de la fonction de score gros-grain

La fonction de score gros-grain VOR montre de moins bonnes performances que la fonction de score atomique POS : seuls 65 des complexes ont au moins un presque-natif parmi les 10 premiers candidats. Même dans le top100 des candidats, seulement 99 des complexes ont au moins un presque-natif. Ceci est dû à plusieurs facteurs. Tout d'abord, on peut remarquer que le nombre de presque-natifs par complexe est drastiquement plus faible : moins de 150 presque-natifs sur les 10 000 structures, pour

plus de la moitié des complexes. En comparaison, aucun complexe n'avait aussi peu de presque-natifs à l'échelle atomique, puisque leur minimum était à 225 presque-natifs et ce nombre dépassait le millier pour 116 des complexes. Ensuite, le nombre de paramètres complexifie l'apprentissage. Nous avons pu voir dans l'étude des poids que de nombreuses informations ont été extraites par la fonction de score : l'encombrement stérique des acides aminés et nucléiques ainsi que la queue de distribution des proportions d'acides aminés et nucléiques à l'interface, notamment. Mais les 210 paramètres ne sont appris que sur moins de 2 500 candidats à chaque fois. De plus, la génération de candidats par perturbation n'a pas permis d'obtenir 30 presque-natifs pour tous les complexes et ne permet qu'une génération inégale du nombre de presque-natifs par complexe. On peut notamment voir des illustrations de complexes pour lesquels la prédiction a fonctionné et d'autres pour lesquels elle n'a pas fonctionné (voir table 4.2).

pdb	ES	TOP10	Attendus	TOP100	Presque-natifs	ROC-AUC
1gtf	1.55	10	6.307	99	6307	0.70
1knz	3.26	10	1.136	90	1136	0.73
1m8v	1.70	10	6.192	92	6192	0.61
2a8v	2.22	10	5.018	90	5018	0.67
2voo	4.88	10	3.261	97	3261	0.77
3d2s	1.63	10	5.557	93	5557	0.62
1asy	2.35	0	0.008	0	8	0.59
1b23	1.67	0	0.024	1	24	0.72
1c0a	1.89	0	0.001	0	1	0.77
1ddl	1.21	0	0.327	1	327	0.61
1di2	0.93	0	0.114	0	114	0.52
1e8o	1.43	0	0.743	10	743	0.60

TABLE 4.2 – Résultats de la fonction de score gros-grain VOR sans contrainte sur l'apprentissage des poids pour 6 des meilleurs (en haut) et 6 des pires (en bas) complexes de la PRIDB (meilleurs ou pires au sens du nombre de presque-natifs dans le top10) : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC sur 10 000 candidats par pdb évalué. On peut remarquer que tous ces meilleurs complexes ont plus de 1 000 presque-natifs dans les 10 000 candidats générés, alors que tous ces pires complexes ont moins de 1 000 presque-natifs générés, souvent même bien moins.

Il est aussi à noter qu'il n'est pas nécessaire d'avoir un maximum de presque-natifs dans le top10 des candidats : un seul presque-natif suffit. De plus, ici, l'objectif est davantage de minimiser la position du premier presque-natif – idéalement pour qu'il soit en première position dans le tri – que de le placer dans le top10. En effet, plus ce presque-natif sera placé loin dans le tri et plus il faudra tester de candidats comme point de départ d'un amarrage atomique avant de pouvoir effectuer un amarrage atomique

sur le bon candidat.

Lorsque l'on regarde le top100 des candidats, on peut observer que plusieurs complexes avec moins de 10 presque-natifs en ont un ou plusieurs dans le top100 (voir table 4.3). Les aires sous la courbe ROC ont alors des valeurs très élevées. On peut donc se rendre compte que, lorsque la génération de candidats par perturbation permet de rapidement générer de nombreux candidats de IRMSD proche de la solution, le protocole d'amarrage gros-grain fonctionne très bien. Mais dès que les deux partenaires sont de taille suffisamment élevée, la génération de 10 000 candidats est insuffisante pour avoir de bonnes chances de trouver l'épitope en une dizaine de tentatives d'amarrage atomique.

pdb	ES	TOP10	Attendus	TOP100	Presque-natifs	ROC-AUC
1ffy	2.30	0	0.008	2	8	0.88
1ser	1.95	0	0.008	1	8	0.88
1qtq	2.05	0	0.007	3	7	0.93
1f7u	2.56	0	0.007	1	7	0.86
1qf6	2.06	0	0.005	1	5	0.85
2bte	2.46	0	0.002	1	2	0.98

TABLE 4.3 – Résultats de la fonction de score gros-grain VOR sans contrainte sur l'apprentissage des poids pour 6 autres des pires complexes de la PRIDB (pires au sens du nombre de presque-natifs dans le top10), mais avec au moins un presque-natif dans le top100 : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC sur 10 000 candidats par pdb évalué. Malgré le faible nombre de presque-natifs dans les 10 000 candidats de chaque complexe, il y a tout de même un ou plusieurs presque-natifs dans le top100 des candidats, ce qui donne de très bonnes ROC-AUC.

4.3.2.3 Comparaison avec contrainte à poids positifs

La fonction de score avec contrainte à poids positifs offre des performances encore moins bonnes. Seulement 8 complexes ont au moins un presque-natif dans le top10 des candidats (voir table 4.4). Et seulement 35 complexes ont au moins un presque-natif dans le top100 des candidats.

Nous pouvons donc voir que la contrainte de l'intervalle de valeurs des poids à apprendre dépend entièrement du contexte de l'apprentissage. Les descripteurs appris sont ici différents du cas de l'échelle atomique et donnent tous, quelle que soit leur nature, une valeur positive ou nulle. La modélisation de la fonction de score doit alors prendre en compte la contrainte selon laquelle tous les poids doivent être positifs, alors que certains termes de score ne sont pas des termes de pénalisation des structures, mais au contraire penchent en faveur de l'interaction.

pdb	ES	TOP10	Attendus	TOP100	Presque-natifs	ROC-AUC
1gtf	1.27	7	6.307	53	6307	0.48
2a8v	0.69	5	5.018	54	5018	0.46
3d2s	1.00	5	5.557	47	5557	0.49
1sds	0.69	4	2.190	15	2190	0.52
1dfu	0.31	1	0.743	2	743	0.31
2bu1	0.35	1	1.291	4	1291	0.38

TABLE 4.4 – Résultats de la fonction de score gros-grain à poids positifs pour les 6 meilleurs complexes de la PRIDB (au sens du nombre de presque-natifs dans le top10) : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC sur 10 000 candidats par pdb évalué. Comme on peut le voir, les résultats ne sont pas satisfaisants pour notre protocole d'amarrage gros-grain.

4.3.2.4 Comparaison avec valeurs de centrage

Étant donné le comportement des structures natives par rapport aux candidats générés, Il est légitime de penser que les presque-natifs ont une plage de valeurs idéales pour au moins certains paramètres, au-dessus et en-dessous de laquelle se trouve un espace majoritairement peuplé de leurres. Nous pouvons tirer parti de cette répartition en plage de valeurs idéales. C'est un concept qui a déjà fait ses preuves, notamment en fouille de données textuelles [214], mais aussi en amarrage protéine-protéine [6]. Une fonction de score non linéaire est alors comparée à la fonction de score constituée d'une combinaison linéaire des termes de score géométriques. Son comportement sur un seul paramètre se modélise par une fonction décroissante jusqu'en 0, suivie d'une fonction croissante (voir fig. 4.16). La valeur de centrage c_i indique le centre estimé par la fonction de score pour la plage de valeurs idéales. Le poids w_i mesure la vitesse à laquelle s'écarter de cette valeur de centrage pénalise la structure. Au lieu d'un seul poids par terme de score i, cette fonction de score non linéaire apprend donc les deux poids w_i et c_i, calculant le score ainsi, avec les valeurs des A descripteurs de l'exemple tels que $X = (x_1, x_2, ...x_{|A|})$ (voir éq. 4.2).

$$f(X) = \sum_{i=1}^{|A|} w_i |x_i - c_i| \tag{4.2}$$

Ce modèle de fonction de score a déjà été présenté au chapitre 3, en section 3.1.1. Son avantage pour le cas qui nous intéresse est de pouvoir discriminer, pour chaque descripteur, d'une part une plage de valeurs correspondant aux presque-natifs, d'autre part des valeurs correspondant aux leurres et positionnées de part et d'autre de cette plage de valeurs. Remarquons que l'apprentissage des poids doit ici se faire dans l'intervalle positif au moins pour les valeurs de centrage. Une valeur de c_i négative n'a en effet pas de sens puisque tous les termes de score sont à valeur dans l'intervalle

positif.

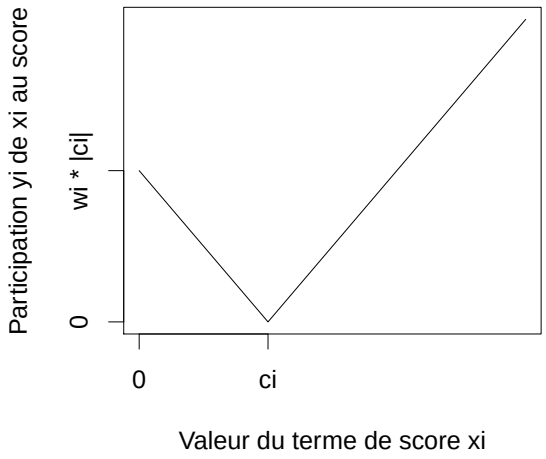


FIGURE 4.16 – Exemple de calcul de la participation y_i à un score y par une fonction de score avec valeurs de centrage, utilisant les poids w_i et c_i. Le poids c_i est aussi appelé la valeur de centrage. On peut voir que l'ordonnée à l'origine est égale à $w_i * |c_i|$. Ce type de fonction permet de modéliser des phénomènes pour lesquels une plage de valeurs idéales est entourée de valeurs de plus en plus éloignées du critère caractérisant la plage de valeurs idéales.

Les résultats de la fonction de score avec valeurs de centrage sont moins satisfaisants (voir table 4.5). Il y a 28 complexes pour lesquels au moins un presque-natif est trouvé dans le top10 des candidats, et 61 complexes avec au moins un presque-natif dans le top100 des candidats. La fonction de score avec valeurs de centrage est tout de même plus performante que la fonction de score formée d'une combinaison linéaire des paramètres dont les poids sont restreints à l'intervalle de valeurs positif. D'autre part, on n'observe plus aucun complexe pour lesquels il existe au moins un presque-natif dans le top100 pour les complexes avec dix presque-natifs ou moins sur l'ensemble des 10 000 candidats générés. Cette fonction de score est donc bien moins adaptée à la prédiction d'interactions protéine-ARN que la fonction de score VOR.

4.4 Conclusions sur la fonction de score gros-grain

Le protocole gros-grain part de candidats générés aléatoirement dans l'espace des candidats pour essayer de trouver un candidat suffisamment proche de l'épitope pour être exploitable par le protocole atomique. Malheureusement, les performances de la fonction de score gros-grain ne remplissent pas aussi bien les objectifs fixés que pour

pdb	ES	TOP10	Attendus	TOP100	Presque-natifs	ROC-AUC
1gtf	1.12	8	6.307	65	6307	0.51
1m8v	1.07	5	6.192	57	6192	0.49
2f8k	1.06	5	3.820	43	3820	0.49
1wsu	1.02	4	2.288	29	2288	0.51
2gxb	0.96	4	3.404	39	3404	0.49
2a8v	0.77	3	5.018	43	5018	0.49

TABLE 4.5 – Résultats de la fonction de score gros-grain avec valeurs de centrage pour les 6 meilleurs complexes de la PRIDB (au sens du nombre de presque-natifs dans le top10) : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC sur 10 000 candidats par pdb évalué. Comme on peut le voir, les résultats sont meilleurs que lorsque l'on contraint un apprentissage des poids dans un intervalle de valeurs positif, mais toujours insatisfaisants pour notre protocole d'amarrage gros-grain.

la fonction de score atomique. Le manque de solutions acceptables à l'échelle gros-grain parmi celles obtenues avec la génération de candidats par perturbation participe à ce constat. Générer plus de 10 000 candidats pourrait permettre de mieux gérer ce phénomène. Le second problème est le nombre de paramètres, qui s'élève à 210, alors que la fonction de score utilise moins de 2 500 candidats dans l'apprentissage. Malgré cela, même pour des complexes avec très peu de solutions, on constate des presque-natifs dans le top100 des candidats.

Nous avons cependant pu constater à nouveau qu'il est important de savoir définir avec précaution l'intervalle de valeurs des poids à apprendre pour la fonction de score. Contrairement à l'amarrage atomique, ici, il convient de ne pas contraindre l'apprentissage des poids à l'intervalle de valeurs positif, au risque d'obtenir une fonction de score moins performante. Nous avons aussi pu tester l'apprentissage d'une fonction de score avec valeurs de centrage, qui semblait bien adaptée au phénomène de plages de valeurs idéales observées pour plusieurs paramètres. Cette fonction de score avec valeurs de centrage était tout de même moins performante que la fonction de score par combinaison linéaire des paramètres.

Nous avons aussi pu montrer que les acides aminés hydrophobes participent bien moins à l'interaction que les acides aminés hydrophiles – surtout les acides aminés chargés – contrairement à ce qui est observé pour les interactions protéine-protéine. Les poids associés aux paramètres issus des volumes médians montrent que la taille (encombrement stérique) des acides aminés et des acides nucléiques est retrouvée et prise en compte dans la fonction de score gros-grain.

Chapitre 5

Discussion biologique

5.1 Limites inhérentes à la construction de fonctions de score obtenues par apprentissage

Les modèles de fonction de score construits pour cette étude sont issus d'extraction de connaissances à partir de données biologiques/biophysiques expérimentales et simulées. Pour donner de bons résultats, ces modèles de fonction de score nécessitent donc des données de qualité et une modélisation fiable et précise des objets étudiés. Or, si la quantité relativement faible de données disponibles (que nous verrons en section 5.1.1) sur les interactions 3D protéine-ARN incite à prédire ces interactions, elle limite aussi les modèles pouvant être construits pour prédire ces interactions.

Pour éviter une sur-représentation de complexes très étudiés dans la PDB, il est nécessaire de limiter la redondance dans les jeux de données. Mais cela a pour conséquence que les quantités de données disponibles sont faibles voire très faibles.

Le manque de données et la diversité des résidus et bases nucléiques imposent des choix de méthodes d'évaluation et de traitement des valeurs manquantes, des valeurs non standards et du solvant. Chacun de ces choix a un impact sur les résultats. Cette partie traite de cet impact, qu'il faut avoir à l'esprit pour correctement interpréter les résultats.

5.1.1 Une source de données expérimentales en constante évolution

Nous avons pu extraire et nettoyer 120 complexes protéines-ARN de la PRIDB non redondante RB199. Nous avons ensuite utilisé ces 120 solutions natives pour générer 10 000 candidats chacune, d'une part pour l'échelle atomique, d'autre part pour l'échelle gros-grain. Pour ces 120 complexes protéine-ARN, nous avons pu montrer que l'objectif était rempli pour plus de 90 % d'entre eux.

Un nouveau jeu de données non redondantes de référence – RB344 – est en cours de construction [8]. Il contiendra comme son nom l'indique 344 chaînes d'acides aminés et palliera en partie le manque de données. Qu'en serait-il si nous avions disposé

8. http://www.pridb.gdcb.iastate.edu/download.php

de ces complexes ? Avec davantage de données, il serait par exemple possible d'envisager une évaluation d'interactions non prises en compte dans le modèle présenté, notamment les interactions avec le solvant explicite qui a été traité lors du nettoyage des données.

5.1.2 Influence du nettoyage des données

5.1.2.1 Valeurs manquantes

Le traitement des valeurs manquantes a ici eu pour but de minimiser l'impact des paramètres inconnus et de permettre d'utiliser des techniques pour lesquelles l'ensemble des descripteurs doivent être renseignés. Dans les jeux de données de référence, certains acides nucléiques (resp. acides aminés) sont peu présents voire inexistants en interaction avec des acides aminés (resp. acides nucléiques). C'est notamment le cas de la cystéine. Pour que le modèle de fonction de score soit complet, il nécessite tout de même un score associé aux interactions rarement rencontrées. Pour déterminer cette valeur de score, nous utilisons une méthode statistique. Une première manière de procéder serait de prendre la valeur moyenne du score pour les autres acides nucléiques ou acides aminés. Ceci donnerait un poids identique à chaque acide nucléique ou aminé dans la valeur du score des interactions rares ou non rencontrées. Or, plusieurs acides nucléiques ou aminés diffèrent de manière importante des autres, que ce soit par leur taille ou par leur charge élevée. Nous avons donc choisi une seconde méthode : la valeur médiane. Pour minimiser l'impact de ces acides aminés ou nucléiques au comportement différent, nous remplaçons la valeur de score des interactions rarement rencontrées par la valeur médiane des acides nucléiques ou aminés sur l'ensemble des structures natives. La médiane de chaque paramètre sur les structures natives suffit pour attribuer une valeur à chacun des 210 paramètres. Il s'ensuit que les interactions rares ou non rencontrées ont une estimation imprécise de la valeur de leur score, mais qui correspond au comportement d'un acide aminé ou nucléique type dans une véritable interaction.

5.1.2.2 Acides aminés et nucléiques non standards

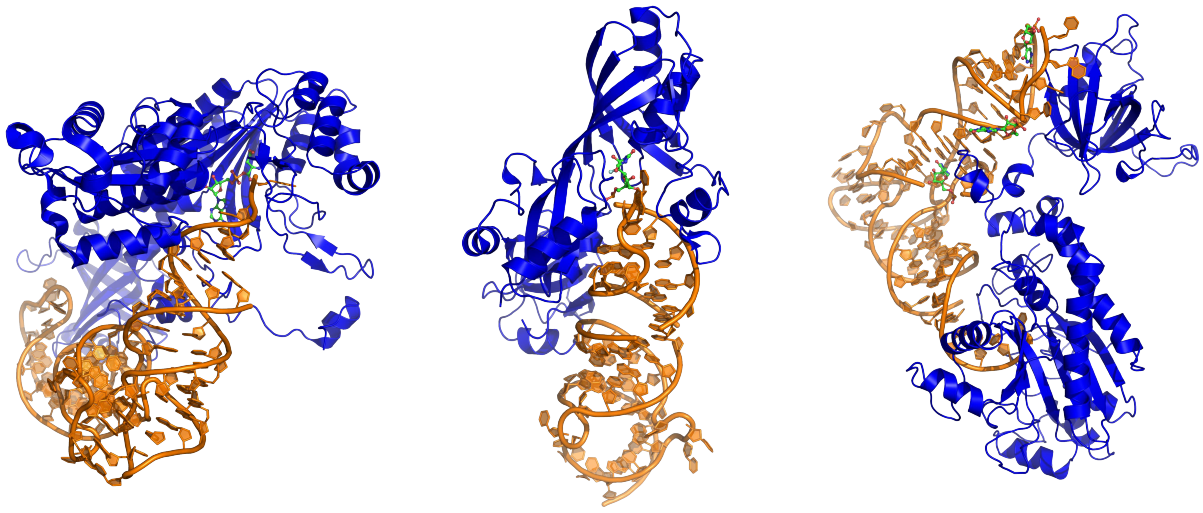
Le traitement des valeurs issues des bases et résidus non standards a un impact sur les prédictions issues du modèle. Pour définir les paramètres de la fonction de score gros-grain, les méthodes utilisées mettent en œuvre des mesures statistiques sur les jeux de données. Le principe est de faire l'hypothèse que toute nouvelle interaction à prédire se comportera localement pour un acide aminé ou un acide nucléique comme les interactions déjà connues :

- si un type d'acide aminé se trouve souvent à l'interaction dans les jeux de données de référence, on aura tendance à penser qu'il devrait généralement être à l'interaction ;
- si un type d'atome ne se trouve quasiment jamais à moins d'une certaine distance d'un autre type d'atome, on aura tendance à penser qu'il ne devrait pas se trouver plus près d'un atome que cette distance.

De plus, il arrive que les protéines et ARN en interaction contiennent des acides aminés ou acides nucléiques non standards, générant des paramètres qui leur sont spécifiques. Or, ces acides nucléiques ou aminés non standards sont rencontrés trop rarement pour modéliser correctement leur influence sur la fonction de score. Il faut alors simplifier le modèle et remplacer les acides aminés et acides nucléiques non standards par la structure chimique standard leur étant la plus proche.

Cette simplification du modèle a peu d'incidence sur la plupart des complexes protéine-ARN. Il existe cependant des cas où les acides aminés ou nucléiques non standards jouent un rôle déterminant dans l'interaction. C'est le cas de certains des complexes mettant en jeu la reconnaissance spécifique de ces acides aminés ou nucléiques non standards. Pour illustration, le complexe tRNA-pseudouridine avec la tRNA-pseudouridine synthase nécessite la reconnaissance spécifique du codon de la pseudouridine du tRNA par la tRNA synthase. Il existe plusieurs complexes de ce type, avec différentes variantes de tRNA-pseudouridine synthase et différents tRNA avec lesquels ils doivent interagir. Pour tous ces complexes, la pseudouridine se trouve clairement à l'interface. Mais la reconnaissance n'est pas garantie lorsque la pseudouridine est remplacée par l'uridine. Commençons par un exemple où le modèle de prédiction admet la reconnaissance du tRNA par la tRNA-synthase même lorsque la pseudouridine est remplacée par l'uridine. Le complexe de code 1r3e existe chez la bactérie *Thermotoga maritima* [193]. Il représente l'interaction entre la tRNA-pseudouridine synthase B et la tRNA-pseudouridine (voir fig. 5.1). Ici, l'interaction est correctement prédite par le modèle et l'on peut facilement observer un entonnoir dans le diagramme d'énergie en fonction du IRMSD (voir fig. 5.2). Mais il est un cas, illustré aussi par la PRIDB, où l'interaction mettant en œuvre des acides nucléiques non standards n'est pas correctement prédite. Ce complexe, de code PDB 1asy, est présent chez *Saccharomyces cerevisiae* [218]. Il s'agit de la tRNA-aspartate synthase A en interaction avec le tRNA-aspartate. Le tRNA-aspartate reconnu contient trois acides nucléiques non standards en différents points de l'interaction, dont la pseudouridine et la méthylguanosine. Notamment, la tRNA-aspartate synthase doit pouvoir différencier le tRNA-aspartate d'un autre tRNA (voir fig. 5.1). L'évaluation du modèle montre que des candidats avec une IRMSD > 5 Å sont présents dans le top10 des candidats en énergie (voir fig. 5.2). La détection d'entonnoir est aussi moins claire. Étant donné le remplacement de la pseudouridine par l'uridine et de la méthylguanosine par la guanine, le modèle propose donc des interactions différentes : la tRNA-aspartate synthase ne doit pas être en interaction avec un tRNA-aspartate dans lequel les acides nucléiques non standards sont remplacés par leur acide nucléique standard.

Certaines molécules biologiques sont aussi considérées comme des hétéroatomes et sont retirées de la structure 3D. C'est notamment le cas des ATP, GTP et autres ligands se liant aux complexes lors de mécanismes qui les mettent en jeu. C'est aussi le cas par exemple de l'aspartyl-adénosine-monophosphate (AMP-ASP). Son implication dans une interaction protéine-ARN est illustrée par le complexe de code 1c0a, présent dans *Escherichia coli* [75]. Ce complexe présente l'interaction de la tRNA-aspartate synthase avec le tRNA-aspartate, mais avec la présence d'un AMP-ASP à l'interaction (voir fig. 5.1). Le vide laissé par cet AMP-ASP n'empêche cependant pas la prédiction de l'interaction (voir fig. 5.2).



a) b) c)

FIGURE 5.1 – Structure 3D de trois complexes protéine-ARN (la protéine en bleu et l'ARN en orange), dans une étude de cas des acides aminés et nucléiques non standards, avec certains acides aminés et nucléiques mis en évidence (bâtonnets verts, rouges et blancs) : a) Le tRNA-aspartate en interaction avec la tRNA-aspartate synthase (code PDB 1c0a), où l'interaction est modélisée et prédite sans difficulté. L'aspartyl-adénosine monophosphate est mise en évidence à l'interaction. b) Le tRNA-pseudouridine en interaction avec la tRNA-pseudouridine synthase (code PDB 1r3e), où l'interaction est aussi correctement prédite. La pseudouridine est mise en évidence à l'interaction. c) Le tRNA-aspartate en interaction avec la tRNA-aspartate synthase de la levure (de code PDB 1asy), avec plusieurs acides nucléiques non standards (PSU – la pseudouridine – et 1MG – la 1N-méthylguanosine) remplacés par les acides nucléiques standards correspondants (respectivement l'uracile et la guanine). Cette interaction est correctement prédite, malgré quelques leurres avec des scores très bas.

5.1.2.3 Solvant et ions

Dans les étapes de nettoyage présentées dans ce manuscrit, le solvant et les ions sont retirés des structures natives. Ce traitement du solvant et des ions permet de simplifier le modèle en retirant le solvant explicite et les ions de la modélisation atomique.

En amarrage protéine-protéine, la communauté CAPRI a mené des évaluations de la prédiction de la position des molécules du solvant et certains algorithmes d'amarrage ont montré de bonnes performances dans l'affinement de la position des molécules d'eau [68]. Cet affinement des positions des molécules d'eau a même permis de mieux modéliser l'interaction entre les deux partenaires.

Dans RosettaDock, les molécules d'eau ne sont pas prises en compte. Elles sont

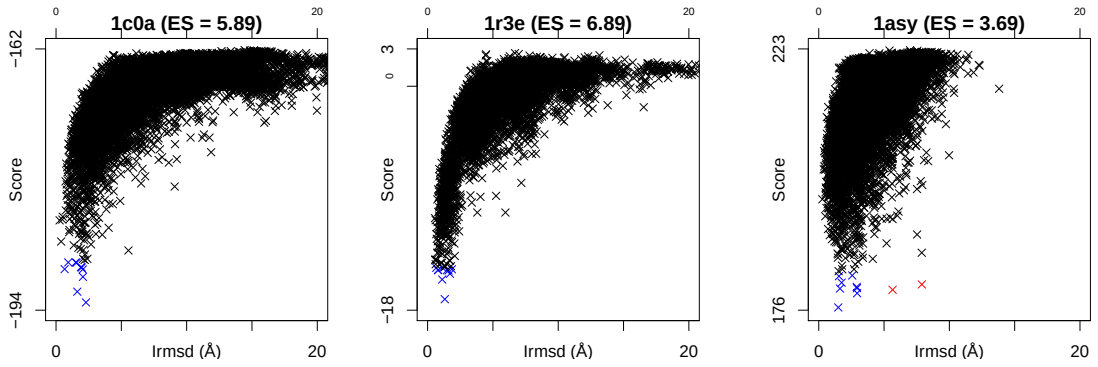


FIGURE 5.2 – Diagramme d'énergie en fonction du IRMSD (EvsRMS) pour les trois complexes de la PRIDB utilisés dans l'étude de cas des partenaires. Pour 1c0a et 1r3e, la prédiction de l'interaction est un succès. Pour 1asy, des améliorations sont encore possibles.

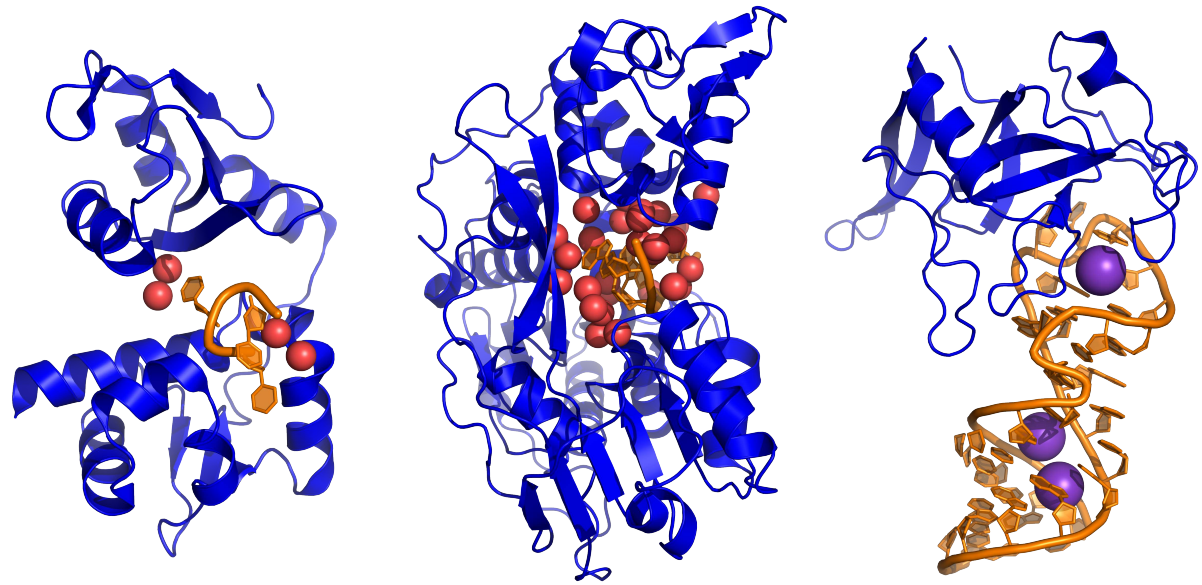
tout simplement ignorées à la lecture du fichier de structure, de même que les ions.

Si l'on devait modéliser l'interaction avec le solvant, il faudrait déterminer les interactions possibles entre chacun des partenaires et chaque molécule d'eau pouvant potentiellement s'incorporer à l'interface. En effet, les molécules d'eau sont des dipôles et peuvent avoir une interaction du côté des hydrogènes comme du côté de l'oxygène. D'une part, modéliser cette interaction revient à traiter un cas plus complexe que le cas de l'interaction binaire. D'autre part, cette énergie est déjà partiellement prise en compte dans le terme de solvatation, qui calcule l'accessibilité au solvant : le solvant est modélisé implicitement [76, 144]. Mais la stabilité conférée par l'interaction indirecte des deux partenaires par le biais de molécules d'eau n'est pas entièrement prise en compte. C'est par exemple le cas du complexe impliquant la protéine *LA* en interaction avec la queue 3' terminale UUU de certains ARN lorsqu'ils sont nouvellement transcrits. C'est le complexe de code PDB 2voo pour la protéine humaine, avec pour rôle la protection de la queue 3'terminale UUU avec laquelle la protéine est en interaction [139]. Dans ce complexe, l'interaction entre les deux partenaires implique de nombreuses molécules d'eau à l'interface, par comparaison avec la taille de l'ARN (voir fig. 5.3). L'évaluation du modèle par l'EvsRMSD montre que la plupart des candidats de plus basse énergie s'éloignent de 3 à 4 Å en IRMSD de la structure native (voir fig. 5.4. La différence est suffisante pour montrer qu'une modélisation plus fine de l'interaction avec les molécules d'eau est nécessaire, au moins pour les complexes dotés d'un partenaire de petite taille.

Un autre exemple de l'interaction avec le solvant peut être observé avec le complexe de code PDB 2jlv du virus de la Dengue [169]. Dans ce complexe, la sous-unité NS3 de la sérine protéase du virus de la Dengue est en interaction avec un petit ARN simple brin. De nombreuses molécules de solvant peuvent être observées entre la sous-unité NS3 et l'ARN, allant jusqu'à quasiment entourer l'ARN (voir fig. 5.3). Comme on peut le voir dans le diagramme EvsRMS, l'interaction de 2jlv n'est pas

prédite correctement par la fonction de score (voir fig. 5.4).

Le cas des ions est illustré par le complexe de code PDB 1jbs d'*Aspergillus restrictus* [263]. La restrictocine d'*Aspergillus restrictus* est en interaction avec un ARN analogue de la boucle du domaine sarcine-ricine de l'ARN 28S. Des ions potassium se trouvent en interaction avec l'ARN et l'un d'entre sépare même l'ARN de la protéine (voir fig. 5.3). La fonction de score a du mal à prédire avec exactitude l'interaction entre la restrictocine et cet analogue de la boucle du domaine sarcine-ricine (voir fig. 5.4).



a) b) c)

FIGURE 5.3 – Structure 3D de trois complexes protéine-ARN (la protéine en bleu et l'ARN en orange), dans une étude de cas du solvant (sphères rouge) et des ions (sphères violettes) : a) La protéine *LA* en interaction avec la queue 3' terminale UUU d'un ARN nouvellement transcrit (code PDB 2voo), avec quatre molécules d'eau assurant l'interaction entre les deux partenaires. Cette interaction est tout de même correctement prédite par la fonction de score. b) La sous-unité NS3 de la sérine protéase du virus de la dengue en interaction avec un petit ARN simple brin (code PDB 2jlv), où l'interaction n'est pas prédite correctement. De nombreuses molécules de solvant sont mises en évidence à l'interface entre les deux partenaires. c) La restrictocine en interaction avec un ARN analogue de la boucle du domaine sarcine-ricine de l'ARN 28S (code PDB 1jbs), où l'on peut voir un ion potassium logé entre la protéine et l'ARN. Cette interaction n'est pas correctement prédite par la fonction de score.

5.1.3 Influence du choix de la méthode d'évaluation

Le *leave-"one-pdb"-out* peut être vu comme une *k*-validation croisée avec *k* égal au nombre de structures natives et où les exemples sont répartis en une strate par

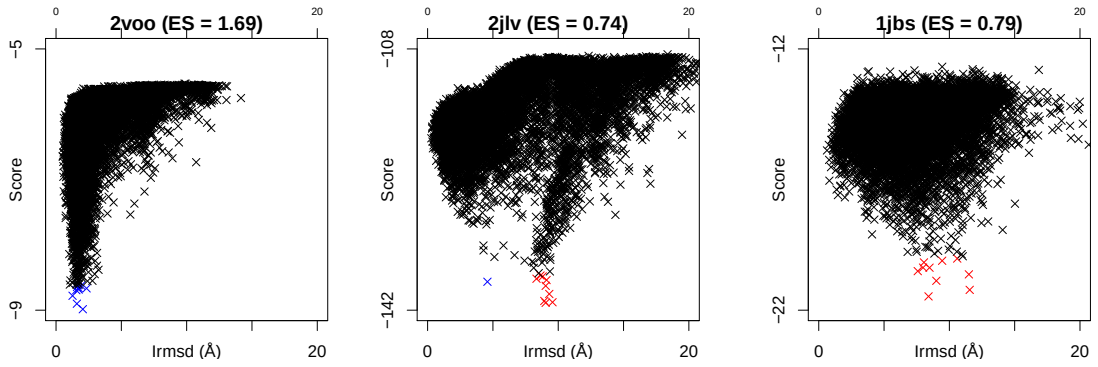


FIGURE 5.4 – Diagramme d'énergie en fonction du IRMSD (EvsRMS) pour les trois complexes de la PRIDB utilisés dans l'étude de cas du solvant et des ions. Pour 2voo et 2jlv, la prédiction de l'interaction est un succès. Pour 1jbs, des améliorations sont encore possibles.

structure native. Chaque strate contient alors les exemples issus de la génération de candidats par perturbation à partir d'une structure native donnée.

Le *leave-"one-pdb"-out* est une méthode permettant de se mettre en situation d'exploitation avec des données d'apprentissage. Plutôt que d'être dans le cas où l'on connaît 120 structures natives, on considère ne connaître que 119 structures natives. On rencontre alors une structure native supplémentaire et l'on évalue la performance de la fonction de score apprise sur cette structure native.

Comme il faut apprendre une fonction de score par strate, les calculs pour l'évaluation des données sont plus coûteux. Ainsi, cette méthode d'évaluation est non seulement préférable pour utiliser un maximum de données en apprentissage, mais son coût évolue aussi en n^2 où n est le nombre de complexes protéine-ARN du jeu de données. Ces deux constats rendent le *leave-"one-pdb"-out* préférable à utiliser sur des jeux de données de petite taille.

Les analyses des résultats sont plus complexes, puisque l'on ne se retrouve plus avec l'évaluation d'une seule fonction de score mais avec l'évaluation de 120 fonctions de score. Il faut s'assurer que les fonctions de score sont toujours comparables et vérifier à quel point les différentes mesures peuvent se comparer entre elles. La ROC-AUC est pour cela une bonne mesure de la performance d'une fonction de score. Pour illustration, le score d'enrichissement à 10 % ne donne pas de résultats satisfaisants, même pour des complexes avec lesquels le diagramme d'EvsIrmsd permet de détecter un entonnoir. Il s'agit généralement de complexes pour lesquels il y a peu de leurres. C'est notamment le cas pour 1av6, 1gtf et 1vfg (voir fig. 5.5). Toutefois, les fonctions de score apprennent à discerner avec un seul en IRMSD < 5 Å un exemple des autres exemples, alors que le score d'enrichissement à 10 % sépare les exemples selon un seuil dépendant de la distribution en IRMSD des exemples.

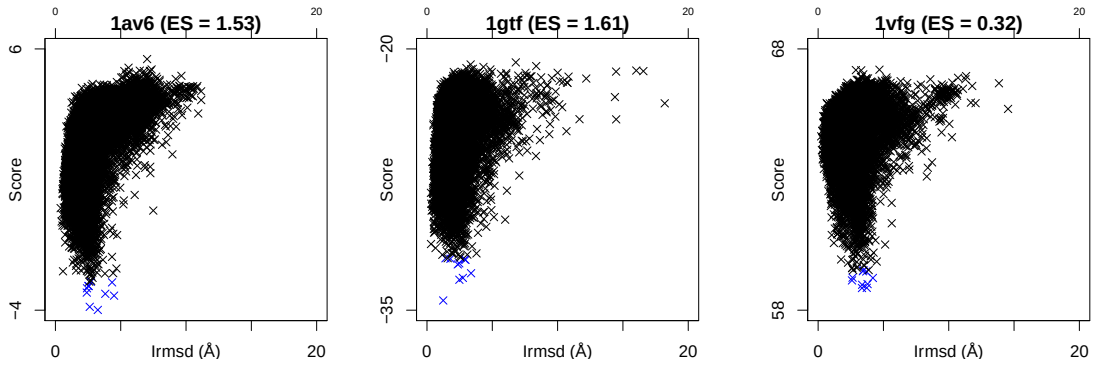


FIGURE 5.5 – Diagramme d'énergie en fonction du IRMSD (EvsRMS) pour trois complexes de la PRIDB pour lesquels on peut observer un faible score d'enrichissement, malgré la détection d'un entonnoir. Il y a peu de leurres pour ces trois complexes, comparés aux autres complexes.

5.2 Flexibilité à l'interaction

Les molécules biologiques adoptent *in vivo* une conformation biologiquement active. C'est cette structure 3D qui est recherchée dans les expériences de repliement et qui est utilisée par l'amarrage pour prédire la structure de l'interaction. Mais il arrive que les structures des protéines et des ARN se déforment à l'interaction. On parle alors de flexibilité.

C'est un concept qui a nécessité de différencier l'amarrage partant des structures liées de l'amarrage partant des structures non liées. En effet, la prédiction de l'interaction est plus difficile avec des structures non liées qu'avec des structures liées. La prédiction est d'autant plus difficile que la déformation de la structure de l'un ou l'autre des deux partenaires est grande à l'interface. Certaines grandes déformations sont modélisables lorsqu'elles mettent en jeu, notamment pour les protéines, des acides aminés se situant dans des boucles.

Il arrive aussi qu'un ARN en interaction avec une protéine ne soit composé que de quelques acides nucléiques. Dans ce genre de cas, la détermination du repliement de l'ARN doit se faire à la volée, directement à l'interface avec la protéine. Si ces calculs peuvent être coûteux pour des ARN de plusieurs dizaines ou centaines d'acides nucléiques, ils sont nécessaires pour les plus petits ARN.

Pour les protéines dans RosettaDock, la flexibilité à l'interaction est modélisée grâce à deux mécanismes :

- la reconstruction des chaînes latérales des acides aminés en interaction avec des acides aminés du partenaire ;
- la reconstruction du squelette des boucles en interaction avec des acides aminés du partenaire.

Pour les chaînes latérales, un échantillonnage des différents rotamères possibles est testé en évaluant le score que chaque rotamère testé obtiendrait s'il remplaçait la

chaîne latérale existante. Un algorithme de Monte-Carlo est utilisé pour choisir quel rotamère doit remplacer la chaîne latérale existante.

Pour les boucles, un algorithme de reconstruction de boucle est utilisé : *Cyclic Coordinate Descent* (CCD, [33]). La boucle à reconstruire est d'abord rompue pour permettre un mouvement plus libre de ses acides aminés. Puis, le squelette de chaque acide aminé est repositionné sans prendre en considération sa chaîne latérale. Les chaînes latérales sont enfin repositionées et, si elles sont en interaction avec des acides aminés du partenaire, sont reconstruites par échantillonnage des rotamères possibles, comme décrit au paragraphe précédent. L'algorithme CCD est utilisé pour refermer la boucle.

L'alliance de ces deux mécanismes de reconstruction de chaînes latérales et de boucles permet de modéliser une part importante de la flexibilité des protéines. Ces mécanismes ont toutefois un coût, qui est contrôlé en ne recherchant pas de manière exhaustive les meilleurs rotamères et squelettes de boucle. En effet, pour l'un comme pour l'autre des mécanismes, le meilleur choix, qu'il s'agisse du rotamère ou du squelette, dépend des autres choix à effectuer pour les rotamères ou squelettes. Une recherche exhaustive impliquerait un calcul combinatoire coûteux. Une heuristique est donc utilisée.

Évidemment, adapter ces deux mécanismes de reconstruction au cas de l'ARN signifie disposer de données conséquentes sur la flexibilité des ARN. D'une part, nous devons disposer d'une base de données de rotamères pour les ARN. Il se trouve qu'une base de données de rotamères d'ARN est disponible dans les fichiers de RosettaDock. Cependant, nous avons pu voir en section 3.2.1 du chapitre 3 que la base de données de rotamères des protéines n'était peut-être pas adaptée aux interactions protéine-ARN. Il n'y a donc aucune raison pour que la base de données de rotamères ARN convienne à la prédiction d'interactions protéine-ARN. Cela signifie qu'il faudrait aussi adapter les rotamères ARN aux interactions protéine-ARN. D'autre part, la reconstruction des boucles a été évaluée pour les acides aminés. Elle nécessite une fonction de score dédiée pour évaluer le meilleur squelette de chaque acide aminé. Il faudrait donc aussi adapter cette fonction de score aux interactions protéine-ARN.

5.3 Limites du protocole de génération des candidats

Le tri des candidats d'une interaction entre macromolécules biologiques implique d'avoir au préalable un ensemble de candidats généré. Cet ensemble de candidats a un impact direct sur les performances du tri. Comme on a pu le voir en section 2.4.4 du chapitre 2, un ensemble de candidats judicieusement filtré permet d'accroître considérablement les performances d'une fonction de score. Ainsi, la génération des candidats doit suivre certains objectifs.

Le premier objectif de la génération des candidats est d'offrir un panel suffisamment large et représentatif de candidats pouvant raisonnablement être considérés comme des candidats de l'interaction : les candidats avec deux partenaires trop éloignés pour interagir ou avec trop d'interpénétration pour représenter une interaction biologique sont des leurres évidents. Cet objectif a cependant des implications s'il est rempli. Nous pouvons par exemple observer pour certains des 120 complexes de la PRIDB que les

candidats générés sont pour plus de 99.9 % d'entre eux répartis de part et d'autre d'un intervalle de plus de 1 Å (voir fig. 5.6). Cet état de fait a deux conséquences.

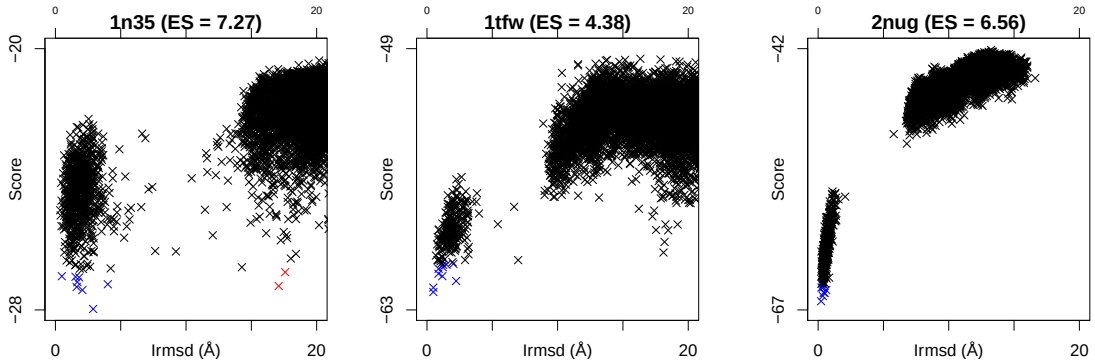


FIGURE 5.6 – Diagramme d'énergie en fonction du IRMSD (EvsRMS) pour trois complexes de la PRIDB pour lesquels on peut observer moins de 10 candidats sur un intervalle de plus de 1 Å, avec le reste des candidats se répartissant de part et d'autre de cet intervalle. Ce manque de données est lié à une barrière énergétique élevée entre les conformations des presque-natifs et celles des leurres.

Tout d'abord, la quasi-absence de candidats dans cet intervalle de IRMSD empêche d'affirmer que, pour ces complexes, il est possible d'affiner une structure proche du natif. Pour tous les autres complexes où un entonnoir est détecté, nous pouvons aisément confirmer que l'affinement de structure est possible. Mais pour ces complexes, comme nous ne disposons pas des candidats dans cet intervalle de IRMSD, nous ne pouvons pas affirmer qu'il n'y aura pas dans cet intervalle un saut en énergie empêchant la formation d'un entonnoir. D'ailleurs, il y a toutes les chances que des candidats générés dans cet intervalle de IRMSD et qui ont été rejetés lors de la génération des candidats aient eu une énergie élevée, notamment due à une très forte contribution du terme de score fa_rep.

Ensuite, il existe une barrière énergétique plus importante pour ces trois complexes que pour les autres entre les conformations des presque-natifs et celles des leurres. Cette barrière énergétique est principalement due à l'interpénétration trop importante entre les deux partenaires, pour les candidats proches des presque-natifs générés. Il s'agit d'ailleurs essentiellement de complexes pour lesquels l'agencement des partenaires en interaction clef-serrure est particulièrement marquée (voir fig. 5.7).

Cette distribution est liée au protocole de génération des candidats par perturbation en corps rigides. Lorsque les molécules sont emboitées, il n'est pas possible d'obtenir un continuum de structures proches et donc un saut apparaît. La fonction de score obtenue pourrait donc avoir de médiocres performances lors de l'évaluation de ce type d'interface par une approche semi-flexible.

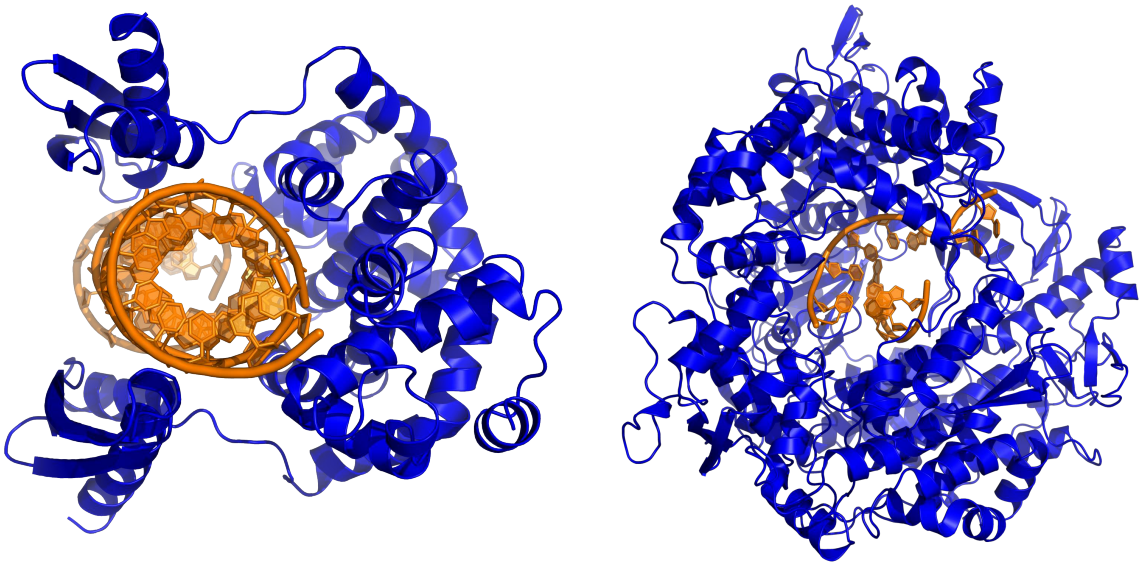


FIGURE 5.7 – Structure 3D d'une interaction de type clef-serrure. On peut voir que la protéine (en bleu) et l'ARN (en orange) s'emboîtent comme le feraient une clef et une serrure.

Chapitre 6

Conclusion et perspectives

6.1 Intégration de connaissances *a priori* de contextes proches

L'amarrage protéine-protéine nous a appris de nombreuses connaissances toujours valables en amarrage protéine-ARN : génération de données à partir d'un jeu de données de référence, termes de score physico-chimiques, contraintes appliquées sur les intervalles de valeurs des poids à apprendre, *etc*. Tout d'abord, nous avons montré que les termes de score physico-chimiques, qui ont fait leur preuve en amarrage protéine-protéine, permettent de capturer des informations décisives dans la prédiction des interactions protéine-ARN. Nous avons montré comment l'apprentissage d'une fonction de score formée d'une combinaison linéaire des paramètres physico-chimiques et apprise grâce à des données de référence nettoyées permet d'atteindre l'objectif fixé. Cet objectif est défini d'après les attentes de la communauté internationale de l'amarrage de macromolécules biologiques, à savoir prédire correctement au moins une structure candidate parmi les 10 meilleures proposées pour chaque interaction. Nous avons pu atteindre cet objectif pour 117 des 120 complexes utilisés pour l'apprentissage de la fonction de score atomique avec ROGER. De plus, pour plus de 90 % des complexes, cette fonction de score montre des capacités à affiner une structure candidate suffisamment proche de la solution, même à 8 Å en IRMSD de la solution. Nous avons aussi montré que cet objectif n'aurait pas été atteint sans l'aide de contraintes appliquées sur les intervalles de valeurs des poids à apprendre pour la fonction de score. Mais il reste des interactions protéine-ARN pour lesquelles les termes de score physico-chimiques sont encore inefficaces, notamment pour ce qui est d'affiner une structure candidate. D'autre part, l'utilisation de ce protocole atomique nécessite d'avoir une idée de l'épitope pour ne pas avoir à générer des millions de candidats.

Nous avons également évalué l'utilisation de données telles que des informations supplémentaires sur le type d'ARN de chacun des complexes protéine-ARN dont l'interaction est à prédire. Nous avons proposé l'apprentissage d'une fonction de score dédiée à chaque catégorie d'ARN parmi trois (ARN simple brin, ARN double brin et ARN de tranfert). Les trois fonctions de score générées ne montrent cependant pas d'amélioration des performances. Il reste encore à tester l'utilisation d'informations

supplémentaires sur les catégories de protéines.

6.2 Extraction des données et complexités des modèles

La modélisation de fonctions de score plus complexes qu'une combinaison linéaire des termes de score physico-chimiques n'a pas permis d'obtenir davantage d'informations sur la prédiction d'interactions protéine-ARN. De nombreux modèles classiques ont été étudiés et il est probable qu'une étude plus approfondie soit nécessaire pour tirer davantage parti des informations obtenues grâce aux termes de score physico-chimiques : apprentissage d'un modèle basé uniquement sur un sous-ensemble des paramètres, apprentissage d'une variable estimée comme l'IRMSD plutôt qu'une classification binaire, *etc*. Des modèles non linéaires tels que des fonctions de score avec valeurs de centrage ou l'utilisation de métaclassifieurs n'a pas non plus apporté de gain de performance. Cependant, il reste possible que les termes de score physico-chimiques aient déjà fourni tout ce qu'on peut en extraire sur l'interaction. Pour améliorer la prédiction, il faudrait alors repenser la modélisation de certains des termes de score physico-chimiques en fonction des différents facteurs d'interaction, pour les adapter aux interactions protéine-ARN. Il n'y a par exemple aucune certitude quant au fait que les critères de décision soient les mêmes pour les interactions protéine-protéine et protéine-ARN entre absence d'interaction (quand les partenaires sont trop éloignés), présence d'interaction (quand les partenaires sont juste à portée) et interpénétration (quand les partenaires sont trop proches l'un de l'autre). L'une des améliorations des termes de score physico-chimiques pourrait donc consister à recalibrer les paramètres propres à la définition de l'interaction et de l'interpénétration pour les interactions protéine-ARN.

Il est cependant concevable d'utiliser des fonctions de score *a posteriori* sur un ensemble plus restreint de candidats, notamment le top10 ou le top100 des candidats du protocole atomique. En ne triant que les meilleurs candidats, il devient possible pour les fonctions de score *a posteriori* de se focaliser sur la discrimination des presque-natifs au sein des candidats les plus proches d'une interaction biologique. On pourrait par exemple attendre des fonctions de score *a posteriori* qu'elles assurent que les presque-natifs du top10 d'un complexe aient plus de chances de se trouver parmi les premiers candidats du top10. Il serait aussi envisageable d'explorer l'hypothèse selon laquelle ces ces fonctions de score tenteraient de retrouver les presque-natifs du top100 et absents du top10 pour les ramener dans le top10.

L'étude *a posteriori*, en n'améliorant pas la prédiction apportée par la combinaison linéaire, nous montre l'importance d'une méthode d'apprentissage adaptée à la problématique posée. De plus, nous avons défini comme objectif pour l'apprentissage par ROGER la maximisation de l'aire sous la courbe ROC. Or, notre objectif véritable n'est pas de maximiser l'aire sous la courbe ROC, mais de maximiser le nombre de presque-natifs dans les 10 premiers candidats du tri. Et pourtant, malgré cette différence entre l'objectif appris et l'objectif véritable à atteindre, cet objectif véritable est rempli. En effet, ne pas utiliser l'objectif réel comme objectif appris est un choix permettant de vérifier si l'apprentissage a su retrouver les informations es-

sentielles pour caractériser les interactions. Plus que cela, les candidats de l'apprentissage étaient seulement étiquetés presque-natifs et leurres, ramenant l'information numérique prodiguée par le IRMSD à une valeur binaire. Rien n'attestait qu'un apprentissage réussi permettrait d'obtenir une fonction quasi-croissante du score en fonction du IRMSD. Ceci montre bien que le modèle de fonction de score formé d'une combinaison linéaire des paramètres physico-chimiques a véritablement pu retrouver les informations déterminantes pour prédire l'interaction protéine-ARN.

Nous avons vu par ailleurs que l'apprentissage de fonctions de score *a posteriori* a confirmé l'importance de certains termes de score dans la description des interactions protéine-ARN et en a révélé de nouveaux. En comparaison avec les interactions protéine-protéine, le facteur le plus important est sans doute le terme de score des forces électrostatiques. Avec une telle importance de ce facteur, une modélisation plus fine des interactions électrostatiques devrait permettre de mieux modéliser l'interaction.

6.3 Prédictions multi-échelle

Nous avons pu montrer qu'une adaptation judicieuse des récentes méthodes d'amarrage protéine-protéine permet de correctement prédire les interactions protéine-ARN. La mise à contribution de termes de score géométriques montre que d'autres voies que celles suggérées par les connaissances issues des scores physico-chimiques peuvent mener à la prédiction des interactions protéine-ARN. Cette vision de l'interaction permet notamment de s'abstraire des seuils de distance utilisés pour définir si deux atomes sont en interaction. Nous avons aussi pu voir que c'est une modélisation qui prend mieux en compte, directement dans la représentation géométrique, la flexibilité de l'ARN, plus accentuée que celle des protéines.

L'apprentissage de la fonction de score gros-grain a nécessité, pour améliorer les performances, de ne pas contraindre l'intervalle de valeurs des poids à apprendre. Simplement contraindre l'apprentissage à l'intervalle de valeurs positif a en effet montré des performances dégradées. Cependant, après une analyse détaillée des valeurs des paramètres sur les structures natives, il peut paraître opportun d'imposer des intervalles de valeurs différents pour chacun des poids, en fonction de leur nature. Exactement comme pour les termes de score physico-chimiques, qui ont des valeurs négatives lorsqu'elles favorisent l'interaction et positives lorsqu'elles la pénalisent, nous pourrions donner un poids négatif aux termes de score privilégiant l'interaction et positif aux termes de score la défavorisant. Cette modification du modèle de fonction de score gros-grain VOR pourrait certainement donner lieu à une amélioration de ses performances.

La modélisation et l'évaluation du protocole gros-grain nous a permis de mettre en lumière des connaissances nouvelles sur les interactions protéine-ARN. Contrairement aux interactions protéine-protéine, qui favorisent les acides aminés hydrophobes à l'interaction, ce sont les acides aminés hydrophiles qui sont préférentiellement en interaction avec les acides aminés. Nous avons aussi pu voir que, des quatre acides nucléiques, c'est l'uracile qui est majoritairement présent à l'interaction. Le volume

médian des cellules de Voronoï des acides aminés et nucléiques a aussi permis de retrouver un ordonnancement des acides aminés, d'une part, et des acides nucléiques, d'autre part, en fonction de leur taille (leur encombrement stérique). Les paramètres associés au volume médian permettent de mesurer l'empilement stérique.

Nous avons enfin pu voir que l'apprentissage multi-échelle repose essentiellement sur le changement de point de vue sur l'objet à apprendre et sur le changement d'objectif. Du point de vue de l'apprentissage, changer de point de vue correspond à changer les attributs utilisés, ici des termes de score. S'aider d'attributs de natures différentes pour la prédiction – ici physico-chimiques et géométriques – permet de capturer des informations sinon difficilement accessibles, potentiellement plus adaptées au problème posé pour chacune des échelles. Changer d'objectif pourrait revenir à changer la fonction objectif de l'apprentissage, qui a été l'aire sour la courbe ROC chaque fois que nous avons appris une fonction de score à l'aide de ROGER. Se pose alors la question de savoir si changer la fonction objectif utilisée pour l'apprentissage permettrait d'améliorer les performances de la fonction de score gros-grain. En effet, il suffit qu'un seul presque-natif soit de rang le plus petit pour que la fonction de score gros-grain .

Toutefois, il reste encore des voies à explorer. Le protocole proposé et évalué dans ce manuscrit est décomposé en deux étapes : amarrage gros-grain, puis amarrage atomique. Il est toujours possible d'étudier des protocoles plus sophistiqués d'amarrage, qui permettront de traquer des épitopes potentiels, à une échelle gros-grain, de les explorer à une échelle atomique et de revenir à l'échelle gros-grain si les candidats générés ne sont pas satisfaisants. Cet aller-retour entre amarrage gros-grain et amarrage atomique tire parti des deux perspectives et nécessite d'étudier le meilleur moyen de les faire communiquer dans trois buts :

- minimiser le nombre d'amarrages atomiques explorés ;
- maximiser les chances de repérer l'épitope ;
- maximiser les chances de trouver un presque-natif une fois l'épitope repéré.

Si l'on paramètre les amarrages gros-grain et atomiques pour qu'ils prennent chacun approximativement le même temps d'exécution, chaque erreur de l'amarrage gros-grain conduisant à un amarrage atomique inutile augmente considérablement les temps de calcul nécessaires avant de trouver la solution. D'un autre côté, si l'amarrage gros-grain ou l'amarrage atomique manquent l'interaction à son échelle, c'est tout le protocole d'amarrage qui est compromis. Un tel protocole pourrait bénéficier d'une fonction de score permettant de définir s'il y a ou non interaction. Jusqu'à maintenant, les fonctions de score définies ici se contentent de trier les candidats et de proposer les meilleurs d'entre eux pour représenter l'interaction, même si l'ensemble des candidats utilisés dans le tri sont des leurres évidents. Or, pour juger que les candidats proposés par un amarrage atomique sont insatisfaisants et ainsi remettre en cause une solution proposée par l'amarrage gros-grain, il est nécessaire de pouvoir rejeter un candidat jugé trop éloigné de ce que devrait être une interaction protéine-ARN. Cela fait appel à une notion de seuil allant au-delà de l'approche par tri adoptée dans ce manuscrit. La fonction de tri, associée au seuil du top10 des candidats, ne fait que sélectionner 10 candidats parmi les 10 000 générés et les propose dans un certain ordre. Cette fonction de tri n'a actuellement aucun moyen de repérer que l'un des candidats proposés

ne devrait pas faire partie de la liste de 10 candidats. C'est donc une autre approche à mettre en œuvre, pouvant par exemple évaluer le nombre de leurres en queue de distribution d'un tri pour s'assurer de pouvoir rejeter les candidats étant des leurres. Mais ce n'est pas suffisant. Pour obtenir un seuil fixe et universel pour l'ensemble des complexes protéine-ARN, il faudrait modéliser une fonction de score dont l'amplitude du score ne dépend pas de la taille du complexe (en nombre d'atomes). En effet, le score obtenu avec les fonctions de score présentées dans ce manuscrit, utilisées dans RosettaDock et dans plusieurs autres algorithmes d'amarrage, est calculé en faisant une somme sur l'ensemble des atomes ou acides aminés et nucléiques. Un complexe de très petite taille aura donc de grandes chances d'avoir un score de faible amplitude alors qu'un complexe de grande taille aura de grandes chances d'avoir un score de forte amplitude. Avec de tels écarts entre les scores, il est impossible de fixer un seuil pour l'ensemble des complexes protéine-ARN qui permette de déterminer à partir de quand un candidat doté de ce score est nécessairement un presque-natif ou un leurre. Il est donc nécessaire, pour traiter du problème de variabilité du seuil, de traiter du problème de la variabilité de taille des objets étudiés et de son impact sur le score. La fonction de score gros-grain résoud en partie ce problème, en proposant quatre types de paramètres indépendants de la taille du complexe (proportions et volumes médians des acides aminés et nucléiques à l'interface, proportions et distances médianes des paires d'acides aminés et nucléiques à l'interface) et deux types de paramètres qui en dépendent (nombre d'acides aminés et nucléiques à l'interface). Pour cette fonction de score, il serait possible de se soustraire totalement de la taille du complexe en modifiant ces deux derniers types de paramètres en mesurant par exemple à la place le pourcentage d'acides aminés et nucléiques à l'interface et la surface médiane d'une facette de l'interface. Il est aussi envisageable d'étudier, en fonction de la taille des complexes protéine-ARN, l'évolution du nombre d'acides aminés et nucléiques à l'interface, pour se faire une idée de leur courbe d'évolution l'un par rapport à l'autre. Cette courbe n'est pas nécessairement linéaire et pourrait conduire à dresser des catégories de complexes en fonction de leur taille.

Plus qu'un protocole multi-échelle de prédiction d'interactions protéine-ARN, c'est une approche que nous avons conçue de sorte qu'elle est adaptable : d'autres protocoles d'amarrage peuvent voir le jour en adaptant cette approche à d'autres algorithmes de génération de candidats et d'autres termes de score. Et plus généralement, la mise au point d'une méthodologie telle que le *leave-"one-pdb"-out* peut parfaitement être réutilisée dans d'autes contextes informatiques. Le cadre le plus adapté au *leave-"one-pdb"-out* est certainement la faible quantité d'instances positives connues, coûteuses ou rares à obtenir, pour lesquelles on souhaite apprendre, étant donné un ensemble de paramètres connus sur ces instances positives, à les retrouver. Il peut s'agir d'apprendre à les modéliser pour les reproduire (commande souhaitée accomplie pour la manipulation d'une interface neuronale) ou pour les éviter (accident en vol pour un avion ou un drone). Cette méthodologie nécessite une manière, à partir de ces instances positives, de générer des exemples proches de ces instances positives et des exemples plus éloignés. Elle implique aussi de disposer d'une mesure de distance ou tout au moins de divergeance entre les exemples et l'instance positive dont ils sont issus. Il reste ensuite à apprendre la donnée recherchée en fonction des

paramètres connus, en veillant à chaque fois à garder pour l'évaluation les exemples issus d'une même instance positive.

Au-delà des protocoles d'amarrage se pose la question du filtrage collaboratif. Jusqu'ici, nous n'avons émis l'hypothèse d'un filtrage collaboratif que pour les termes de score physico-chimiques et les scores issus des prédictions des modèles *a posteriori*. Cependant, nous pouvons toujours étudier les filtrages collaboratifs dans le cadre de l'échelle gros-grain, potentiellement en conjonction avec les termes de score physico-chimiques de l'échelle gros-grain.

Nous avions envisagé dans la section précédente une autre façon de traiter l'étude *a posteriori*, en lui donnant un but différent de la prédiction par combinaison linéaire. Puisqu'il s'agit d'un tri *a posteriori*, le tri par combinaison linéaire est déjà effectué lorsque le tri *a posteriori* est appliqué. Dans ce cas de figure, il est tout à fait possible de n'appliquer le tri *a posteriori* que sur un sous-ensemble déjà trié des prédictions. Comme le tri par combinaison linéaire remplit les objectifs fixés pour presque tous les complexes, il est pensable de durcir les objectifs en intégrant le tri *a posteriori* pour ne trier que les 10 ou 100 premiers candidats. De la sorte, le tri *a posteriori* ne traite que des candidats déjà jugés satisfaisants et pourrait être en mesure de donner au moins un presque-natif dans les trois, quatre ou cinq premiers candidats. Si un tel objectif peut être atteint, cela signifie qu'il est possible, toujours en proposant 10 candidats potentiels de l'interaction, de choisir des candidats représentant le mieux l'interaction telle qu'elle est vue par différents amarrages atomiques. Chacun de ces amarrages atomiques serait initié par un candidat gros-grain différent et ferait l'hypothèse d'un épitope distinct.

L'amarrage gros-grain tel qu'il est modélisé dans ce manuscrit met en œuvre 210 paramètres dans sa fonction de score. De plus, 104 de ces paramètres ont, pour la plupart des candidats, une valeur non attribuée et remplacée par la valeur médiane du paramètre sur l'ensemble des structures natives. Le premier facteur à l'origine de ces deux constatations est que nous différencions les vingt types d'acides aminés et les quatre types d'acides nucléiques. En effet, 208 des paramètres sont définis par le nombre de types d'acides aminés et d'acides nucléiques pris en compte : le nombre de certains de ces paramètres est issu d'une addition du nombre de types d'acides aminés et du nombre de types d'acides nucléiques ; pour d'autres de ces paramètres, il s'agit d'une multiplication. Or, nous avons vu que des résultats satisfaisants en amarrage protéine-protéine ont été observés en regroupant les acides aminés en six catégories [15]. Une manière de diminuer la complexité du modèle serait donc de reprendre ces six catégories, ramenant d'une part le nombre de paramètres à 64 et diminuant d'autre part le nombre de valeurs non attribuées pour chaque candidat. Confronter ce modèle au modèle à vingt types d'acides aminés permettrait de savoir, dans le cas où certaines informations échappent au modèle à six catégories d'acides aminés, si les catégories d'acides aminés définies pour l'amarrage protéine-protéine peuvent être remaniées pour l'amarrage protéine-ARN. Il a toutefois fallu déjà disposer du protocole tel que nous l'avons défini jusqu'à maintenant pour pouvoir confronter ce nouveau modèle de fonction de score à celui que nous avons évalué.

Les différentes approches évaluées ont donc montré que, rien qu'à partir de la structure 3D de chacun des deux partenaires d'un complexe et sans aucune autre

forme de données, il est possible de retrouver l'interaction de ces deux partenaires. Il faudra certes encore prendre en compte la flexibilité des partenaires pour véritablement passer de la structure 3D de partenaires non liés – dont la structure est déterminée en dehors de toute interaction – à la prédiction de leur interaction. Mais la rapidité de calculs permet désormais de se pencher sur la question de l'amarrage haut débit. La prise en compte de la flexibilité et l'accession à l'amarrage haut-débit constituent certainement la suite logique de ces travaux.

La prédiction d'interactions protéine-ARN est un domaine en plein essor, pour lequel de plus en plus de défis sont relevés. Cela a commencé par la prédiction d'interactions de petites molécules, puis de molécules de plus grandes tailles. La prédiction d'interactions dans le cas où la protéine est non liée est en passe d'être résolue. Nous devrions voir bientôt des travaux traitant de la flexibilité de l'ARN sur le point de résoudre les interactions avec les deux partenaires non liés. Mais ceci nécessitera pour les petits ARN de savoir reconstruire l'ARN à la volée, en prenant en compte la proximité de la protéine. Plus généralement dans le domaine des interactions de macromolécules biologiques, CAPRI a récemment montré sa volonté d'évaluer la prédiction de caractéristiques spécifiques de l'interaction, telles que la position des molécules de solvant ou l'affinité de l'interaction [68]. Avec des fonctions de score mieux adaptées à l'estimation de la valeur d'une énergie ou de l'affinité d'une interaction, il deviendra possible de passer le cap de la classification binaire. Mais de tels défis nécessitent des données biologiques rarement disponibles en grandes quantités, et même souvent disponibles en bien moindres quantités que les structures 3D des interactions. Il faudra encore quelques années avant que l'on puisse utiliser des méthodes basées sur la connaissance pour prédire l'affinité de l'interaction, et encore plus pour les complexes protéine-ARN.

Bibliographie

[1] L. Adamian, R. Jackups, Jr, T. A. Binkowski, and J. Liang. Higher-order interhelical spatial interactions in membrane proteins. *J Mol Biol*, 327(1) :251–72, 2003.

[2] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. In *Machine Learning*, pages 37–66, 1991.

[3] B. Angelov, J. F. Sadoc, R. Jullien, A. Soyer, J. P. Mornon, and J. Chomilier. Nonatomic solvent-driven Voronoi tessellation of proteins : an open tool to analyze protein folds. *Proteins*, 49(4) :446–56, 2002.

[4] J. C. Austin, K. R. Rodgers, and T. G. Spiro. Protein structure from ultraviolet resonance Raman spectroscopy. *Methods Enzymol*, 226 :374–96, 1993.

[5] G. S. Ayton, W. G. Noid, and G. A. Voth. Multiscale modeling of biomolecular systems : in serial and in parallel. *Curr Opin Struct Biol*, 17(2) :192–198, Apr 2007.

[6] J. Azé, T. Bourquard, S. Hamel, A. Poupon, and D. Ritchie. *Using Kendall-tau Meta-Bagging to Improve Protein-Protein Docking Predictions*, volume 7036 of *Lecture Notes in Computer Science*, chapter 25, pages 284–295. Springer Berlin Heidelberg, 2011.

[7] R. P. Bahadur, P. Chakrabarti, F. Rodier, and J. Janin. A dissection of specific and non-specific protein-protein interfaces. *J Mol Biol*, 336(4) :943–55, 2004.

[8] D. Bakowies and W. F. Van Gunsteren. Water in protein cavities : a procedure to identify internal water and exchange pathways and application to fatty acid-binding protein. *Proteins*, 47(4) :534–45, 2002.

[9] A. Barik, N. C., and R. P. Bahadur. A protein-RNA docking benchmark (I) : nonredundant cases. *Proteins*, 80(7) :1866–1871, Jul 2012.

[10] E. Ben-Zeev, A. Berchanski, A. Heifetz, B. Shapira, and M. Eisenstein. Prediction of the unknown : inspiring experience with the CAPRI experiment. *Proteins*, 52(1) :41–6, 2003.

[11] E. Ben-Zeev and M. Eisenstein. Weighted geometric docking : incorporating external information in the rotation-translation scan. *Proteins*, 52(1) :24–7, 2003.

[12] A. Berchanski, D. Segal, and M. Eisenstein. Modeling oligomers with Cn or Dn symmetry : application to CAPRI target 10. *Proteins*, 60(2) :202–6, 2005.

[13] H. M. Berman, T. N. Bhat, P. E. Bourne, Z. Feng, G. Gilliland, H. Weissig, and J. Westbrook. The Protein Data Bank and the challenge of structural genomics. *Nat Struct Biol*, 7 Suppl :957–9, 2000.

[14] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28 :235–242, 2000.

[15] J. Bernauer, J. Azé, J. Janin, and A. Poupon. A new protein-protein docking scoring function based on interface residue properties. *Bioinformatics*, 23(5) :555–562, Mar 2007.

[16] J. Bernauer, R. P. Bahadur, F. Rodier, J. Janin, and A. Poupon. DiMoVo : a Voronoi tessellation-based method for discriminating crystallographic and biological protein-protein interactions. *Bioinformatics*, 24(5) :652–658, Mar 2008.

[17] J. Bernauer, X. Huang, A. Y. Sim, and M. Levitt. Fully differentiable coarse-grained and all-atom knowledge-based potentials for RNA structure evaluation. *RNA*, 17(6) :1066–75, 2011.

[18] J. Bernauer, A. Poupon, J. Azé, and J. Janin. A docking analysis of the statistical physics of protein-protein recognition. *Phys Biol*, 2(2) :S17–23, 2005.

[19] T. A. Binkowski, L. Adamian, and J. Liang. Inferring functional relationships of proteins from local sequence and spatial surface patterns. *J Mol Biol*, 332(2) :505–26, 2003.

[20] D. M. Blow, C. S. Wright, D. Kukla, A. Ruhlmann, W. Steigemann, and R. Huber. A model for the association of bovine pancreatic trypsin inhibitor with chymotrypsin and trypsin. *J Mol Biol*, 69(1) :137–44, 1972.

[21] A. Bondi. Van der Waals volumes and radii. *The Journal of physical chemistry*, 68(3) :441–451, 1964.

[22] A. J. Bordner and A. A. Gorin. Protein docking using surface matching and supervised machine learning. *Proteins*, 68(2) :488–502, 2007.

[23] D. Bostick and I. I. Vaisman. A new topological method to measure protein structure similarity. *Biochem Biophys Res Commun*, 304(2) :320–5, 2003.

[24] L. G. Boulu, G. M. Crippen, H. A. Barton, H. Kwon, and M. A. Marletta. Voronoi binding site model of a polycyclic aromatic hydrocarbon binding protein. *J Med Chem*, 33(2) :771–5, 1990.

[25] P. E. Bourne. CASP and CAFASP experiments and their findings. *Methods Biochem Anal*, 44 :501–7, 2003.

[26] T. Bourquard, J. Bernauer, J. Aze, and A. Poupon. Comparing voronoi and laguerre tessellations in the protein-protein docking context. In *Voronoi Diagrams, 2009. ISVD '09. Sixth International Symposium on*, pages 225–232, 2009.

[27] T. Bourquard, J. Bernauer, J. Azé, and A. Poupon. A collaborative filtering approach for protein-protein docking scoring functions. *PLoS One*, 6(4) :e18541, 2011.

[28] M. P. Bradley and G. M. Crippen. Voronoi modeling : the binding of triazines and pyrimidines to L. casei dihydrofolate reductase. *J Med Chem*, 36(21) :3171–7, 1993.

[29] P. Bradley, L. Malmstrom, B. Qian, J. Schonbrun, D. Chivian, D. E. Kim, J. Meiler, K. M. Misura, and D. Baker. Free modeling with Rosetta in CASP6. *Proteins*, 61 Suppl 7 :128–34, 2005.

[30] L. Breiman. Random Forests. *Eighteenth International Conference on Machine Learning*, 45(1) :5–32, 2001.

[31] N. Calimet, M. Schaefer, and T. Simonson. Protein molecular dynamics with the generalized Born/ACE solvent model. *Proteins*, 45(2) :144–58, 2001.

[32] C. J. Camacho. Modeling side-chains using molecular dynamics improve recognition of binding region in CAPRI targets. *Proteins*, 60(2) :245–51, 2005.

[33] A. A. Canutescu and R. L. Dunbrack. Cyclic coordinate descent : a robotics algorithm for protein loop closure. *Protein science*, 12(5) :963–972, 2003.

[34] C. W. Carter, Jr, B. C. LeFebvre, S. A. Cammer, A. Tropsha, and M. H. Edgell. Four-body potentials reveal protein-specific correlations to stability changes caused by hydrophobic core mutations. *J Mol Biol*, 311(4) :625–38, 2001.

[35] P. Carter, V. I. Lesk, S. A. Islam, and M. J. Sternberg. Protein-protein docking using 3D-Dock in rounds 3, 4, and 5 of CAPRI. *Proteins*, 60(2) :281–8, 2005.

[36] Project CGAL. CGAL, Computational Geometry Algorithms Library, 1990.

[37] S. Chakravarty, A. Bhinge, and R. Varadarajan. A procedure for detection and quantitation of cavity volumes proteins. Application to measure the strength of the hydrophobic driving force in protein folding. *J Biol Chem*, 277(35) :31345–53, 2002.

[38] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines (version 2.31), 2001.

[39] R. Chen, L. Li, and Z. Weng. ZDOCK : an initial-stage protein-docking algorithm. *Proteins*, 52(1) :80–7, 2003.

[40] R. Chen, W. Tong, J. Mintseris, L. Li, and Z. Weng. ZDOCK predictions for the CAPRI challenge. *Proteins*, 52(1) :68–73, 2003.

[41] R. Chen and Z. Weng. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins*, 47(3) :281–94, 2002.

[42] R. Chen and Z. Weng. A novel shape complementarity scoring function for protein-protein docking. *Proteins*, 51(3) :397–408, 2003.

[43] Y. Chen, T. Kortemme, T. Robertson, D. Baker, and G. Varani. A new hydrogen-bonding potential for the design of protein-RNA interactions predicts specific contacts and discriminates decoys. *Nucleic Acids Res*, 32(17) :5147–62, 2004.

[44] Y. Chen and G. Varani. Protein families and RNA recognition. *FEBS J*, 272(9) :2088–97, 2005.

[45] Y. Chen and G. Varani. Engineering RNA-binding proteins for biology. *FEBS J*, 280(16) :3734–54, 2013.

[46] J. Cherfils, S. Duquerroy, and J. Janin. Protein-protein recognition analyzed by docking simulation. *Proteins*, 11(4) :271–80, 1991.

[47] C. Chothia and M. Gerstein. Protein evolution. How far can sequences diverge ? *Nature*, 385(6617) :579, 581, 1997.

[48] C. Chothia and J. Janin. Principles of protein-protein recognition. *Nature*, 256(5520) :705–8, 1975.

[49] A. Clery, M. Blatter, and F. H. Allain. RNA recognition motifs : boring ? Not quite. *Curr Opin Struct Biol*, 18(3) :290–8, 2008.

[50] W. W. Cohen. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[51] N. Colloc'h, C. Etchebest, E. Thoreau, B. Henrissat, and J.P. Mornon. Comparison of three algorithms for the assignment of secondary structure in proteins : the advantages of a consensus assignment. *Protein Eng*, 6(4) :377–82, 1993.

[52] S. R. Comeau and C. J. Camacho. Predicting oligomeric assemblies : N-mers a primer. *J Struct Biol*, 150(3) :233–44, 2005.

[53] S. R. Comeau, S. Vajda, and C. J. Camacho. Performance of the first protein docking server ClusPro in CAPRI rounds 3-5. *Proteins*, 60(2) :239–44, 2005.

[54] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5) :548–558, 1983.

[55] M. L. Connolly. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221(4612) :709–713, Aug 1983.

[56] M. L. Connolly. Molecular surface triangulation. *Journal of Applied Crystallography*, 18(6) :499–505, 1985.

[57] M. L. Connolly. Shape complementarity at the hemoglobin alpha 1 beta 1 subunit interface. *Biopolymers*, 25(7) :1229–47, 1986.

[58] M. L. Connolly. Molecular interstitial skeleton. *Computers & Chemistry*, 15(1) :37–45, 1991.

[59] L. L. Conte, C. Chothia, and J. Janin. The atomic structure of protein-protein recognition sites. *J Mol Biol*, 285(5) :2177–98, 1999.

[60] T. A. Cooper, L. Wan, and G. Dreyfuss. RNA and disease. *Cell*, 136(4) :777–93, 2009.

[61] A. Cornuiéjols and L. Miclet. *Apprentissage Artificiel - Concepts et algorithmes*. Cépadues, 2002.

[62] D. Cozzetto, A. Di Matteo, and A. Tramontano. Ten years of predictions... and counting. *FEBS J*, 272(4) :881–2, 2005.

[63] G. M. Crippen. Voronoi binding site models. *NIDA Res Monogr*, 112 :7–20, 1991.

[64] M. D. Daily, D. Masica, A. Sivasubramanian, S. Somarouthu, and J. J. Gray. CAPRI rounds 3-5 reveal promising successes and future challenges for RosettaDock. *Proteins*, 60(2) :181–6, 2005.

[65] R. Das and D. Baker. Automated de novo prediction of native-like RNA tertiary structures. *Proc Natl Acad Sci U S A*, 104(37) :14664–9, 2007.

[66] R. Das, J. Karanicolas, and D. Baker. Atomic accuracy in predicting and designing noncanonical RNA structure. *Nat Methods*, 7(4) :291–4, 2010.

[67] S. J. de Vries, A. S. Melquiond, P. L. Kastritis, E. Karaca, A. Bordogna, M. van Dijk, J. P. Rodrigues, and A. M. Bonvin. Strengths and weaknesses of data-driven docking in critical assessment of prediction of interactions. *Proteins*, 78(15) :3242–9, 2010.

[68] S. J. De Vries, A. D. J. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar, and A. M. J. J. Bonvin. HADDOCK versus HADDOCK : new features and performance of HADDOCK2. 0 on the CAPRI targets. *Proteins : structure, function, and bioinformatics*, 69(4) :726–733, 2007.

[69] C. Dominguez, R. Boelens, and A. M. Bonvin. HADDOCK : a protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc*, 125(7) :1731–7, 2003.

[70] R. L. Dunbrack, Jr and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci*, 6 :1661–1681, 1997.

[71] F. Dupuis, J.F. Sadoc, R. Jullien, B. Angelov, and J.P. Mornon. Voro3D : 3D Voronoi tessellations applied to protein structures. *Bioinformatics*, 21(8) :1715–6, 2005.

[72] H. Edelsbrunner, M. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Pac Symp Biocomput*, pages 272–87, 1996.

[73] H. Edelsbrunner, M. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Discr Appl Math*, 88 :83–102, 1998.

[74] H. Edelsbrunner and P. Koehl. The weighted-volume derivative of a space-filling diagram. *Proc Natl Acad Sci U S A*, 100(5) :2203–8, 2003.

[75] S. Eiler, A.-C. Dock-Bregeon, L. Moulinier, J.-C. Thierry, and D. Moras. Synthesis of aspartyl-tRNAAsp in Escherichia coli—a snapshot of the second step. *The EMBO journal*, 18(22) :6532–6541, 1999.

[76] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319(6050) :199–203, 1986.

[77] M. Eisenstein. Introducing a 4th dimension to protein-protein docking. *Structure (Camb)*, 12(12) :2095–6, 2004.

[78] J. J. Ellis, M. Broom, and S. Jones. Protein-RNA interactions : structural analysis and functional classes. *Proteins*, 66(4) :903–11, 2007.

[79] J. Fernández-Recio, R. Abagyan, and M. Totrov. Improving CAPRI predictions : optimized desolvation for rigid-body docking. *Proteins*, 60(2) :308–13, 2005.

[80] C. Ferri, P. Flach, and J. Hernández-Orallo. Learning decision trees using the area under the ROC curve. *Machine Learning-International Workshop Then Conference-*, pages 139–146, 2002.

[81] S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230) :245–6, 1989.

[82] J. L. Finney. Volume occupation, environment and accessibility in proteins. The problem of the protein surface. *J Mol Biol*, 96(4) :721–32, 1975.

[83] D. Fischer, S. L. Lin, H. L. Wolfson, and R. Nussinov. A geometry-based suite of molecular docking processes. *J Mol Biol*, 248(2) :459–77, 1995.

[84] D. Fischer, R. Norel, H. Wolfson, and R. Nussinov. Surface motifs by a computer vision technique : searches, detection, and implications for protein-ligand recognition. *Proteins*, 16(3) :278–92, 1993.

[85] S. C. Flores, J. Bernauer, S. Shin, R. Zhou, and X. Huang. Multiscale modeling of macromolecular biosystems. *Brief Bioinform*, 13(4) :395–405, 2012.

[86] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. *Fifteenth International Conference on Machine Learning*, 45(1) :144–151, 1998.

[87] D. Frishman and P. Argos. The future of protein secondary structure prediction accuracy. *Fold Des*, 2(3) :159–62, 1997.

[88] J. J. Fritz, A. Lewin, W. Hauswirth, A. Agarwal, M. Grant, and L. Shaw. Development of hammerhead ribozymes to modulate endogenous gene expression for functional studies. *Methods*, 28(2) :276–285, Oct 2002.

[89] J. Fürnkranz and Flach. An analysis of rule evaluation metrics. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Jan 2003.

[90] H. H. Gan, A. Tropsha, and T. Schlick. Lattice protein folding with two and four-body statistical potentials. *Proteins*, 43(2) :161–74, 2001.

[91] E. J. Gardiner, P. Willett, and P. J. Artymiuk. GAPDOCK : a Genetic Algorithm Approach to Protein Docking in CAPRI round 1. *Proteins*, 52(1) :10–4, 2003.

[92] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868) :141–7, 2002.

[93] B. J. Gellatly and J. L. Finney. Calculation of protein volumes : an alternative to the Voronoi procedure. *J Mol Biol*, 161(2) :305–22, 1982.

[94] M. Gerstein and C. Chothia. Packing at the protein-water interface. *Proc Natl Acad Sci U S A*, 93(19) :10167–72, 1996.

[95] M. Gerstein, J. Tsai, and M. Levitt. The volume of atoms on the protein surface : calculated from simulation, using Voronoi polyhedra. *J Mol Biol*, 249(5) :955–66, 1995.

[96] A. Goede, R. Preissner, and C. Frömmel. Voronoi cell : new method for allocation of space among atoms : elimination of avoidable errors in calculation of atomic volume and density. *J Comp Chem*, 18(9) :1113–1123, 1997.

[97] S. C. B. Gopinath. Mapping of RNA-protein interactions. *Anal Chim Acta*, 636(2) :117–128, Mar 2009.

[98] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J Mol Biol*, 331(1) :281–99, 2003.

[99] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J Mol Biol*, 331(1) :281–99, 2003.

[100] J. J. Gray, S. E. Moughon, T. Kortemme, O. Schueler-Furman, K. M. Misura, A. V. Morozov, and D. Baker. Protein-protein docking predictions for the CAPRI experiment. *Proteins*, 52(1) :118–22, 2003.

[101] A. Guilhot-Gaudeffroy, J. Azé, J. Bernauer, and C. Froidevaux. Apprentissage de fonctions de tri pour la prédiction d'interactions protéine-ARN. In *Quatorzième conférence Francophone sur l'Extraction et la Gestion des Connaissances*, pages 479–484, 2014.

[102] D. M. Halaby and J. P. Mornon. The immunoglobulin superfamily : an insight on its tissular, species, and functional diversity. *J Mol Evol*, 46(4) :389–400, 1998.

[103] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software : an update. *SIGKDD Explorations*, 11(1) :10–18, 2009.

[104] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking : an overview of search algorithms and a guide to scoring functions. *Proteins*, 47(4) :409–43, 2002.

[105] Wei Han, Cheuk-Kin Wan, Fan Jiang, and Yun-Dong Wu. Pace force field for protein simulations. 1. full parameterization of version 1 and verification. *Journal of Chemical Theory and Computation*, 6(11) :3373–3389, 2010.

[106] Wei Han, Cheuk-Kin Wan, and Yun-Dong Wu. Pace force field for protein simulations. 2. folding simulations of peptides. *Journal of Chemical Theory and Computation*, 6(11) :3390–3402, 2010.

[107] Y. Harpaz, M. Gerstein, and C. Chothia. Volume changes on protein folding. *Structure*, 2(7) :641–9, 1994.

[108] A. Heifetz, E. Katchalski-Katzir, and M. Eisenstein. Electrostatics in protein-protein docking. *Protein Sci*, 11(3) :571–87, 2002.

[109] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A. R. Willems, H. Sassi, P. A. Nielsen, K. J. Rasmussen, J. R. Andersen, L. E. Johansen, L. H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B. D. Sorensen, J. Matthiesen, R. C. Hendrickson, F. Gleeson, T. Pawson, M. F. Moran, D. Durocher, M. Mann, C. W. Hogue, D. Figeys, and M. Tyers. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868) :180–3, 2002.

[110] I. L. Hofacker. RNA secondary structure analysis using the Vienna RNA package. *Curr Protoc Bioinformatics*, Chapter 12 :Unit 12.2, Feb 2004.

[111] G. M. Huang. High-throughput DNA sequencing : a genomic data manufacturing process. *DNA Seq*, 10(3) :149–53, 1999.

[112] S. Y. Huang and X. Zou. A nonredundant structure dataset for benchmarking protein-RNA computational docking. *J Comput Chem*, 34(4) :311–8, 2013.

[113] S. Y. Huang and X. Zou. A knowledge-based scoring function for protein-RNA interactions derived from a statistical mechanics-based iterative method. *Nucleic Acids Res*, 2014.

[114] Y. Huang, S. Liu, D. Guo, L. Li, and Y. Xiao. A novel protocol for three-dimensional structure prediction of RNA-protein complexes. *Scientific reports*, 3, 2013.

[115] Y. Inbar, D. Schneidman-Duhovny, I. Halperin, A. Oron, R. Nussinov, and H. J. Wolfson. Approaching the CAPRI challenge with an efficient geometry-based docking. *Proteins*, 60(2) :217–23, 2005.

[116] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast : a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci U S A*, 97(3) :1143–7, 2000.

[117] S. Izvekov and G. A. Voth. A multiscale coarse-graining method for biomolecular systems. *J Phys Chem B*, 109(7) :2469–2473, Feb 2005.

[118] J. Janin. Assessing predictions of protein-protein interaction : the CAPRI experiment. *Protein Sci*, 14(2) :278–83, 2005.

[119] J. Janin. Sailing the route from Gaeta, Italy, to CAPRI. *Proteins*, 60(2) :149, 2005.

[120] J. Janin. The targets of CAPRI rounds 3-5. *Proteins*, 60(2) :170–5, 2005.

[121] J. Janin. Protein-protein docking tested in blind predictions : the CAPRI experiment. *Mol Biosyst*, 6(12) :2351–62, 2010.

[122] J. Janin, K. Henrick, J. Moult, L. T. Eyck, M. J. Sternberg, S. Vajda, I. Vakser, and S. J. Wodak. CAPRI : a Critical Assessment of PRedicted Interactions. *Proteins*, 52(1) :2–9, 2003.

[123] J. Janin and B. Seraphin. Genome-wide studies of protein-protein interaction. *Curr Opin Struct Biol*, 13(3) :383–8, 2003.

[124] J. Janin and S. J. Wodak. Reaction pathway for the quaternary structure change in hemoglobin. *Biopolymers*, 24(3) :509–26, 1985.

[125] J. Janin and S. J. Wodak. Protein modules and protein-protein interaction. Introduction. *Adv Protein Chem*, 61 :1–8, 2002.

[126] F. Jiang and S. H. Kim. "Soft docking" : matching of molecular surface cubes. *J Mol Biol*, 219(1) :79–102, 1991.

[127] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[128] W. Kabsch and C. Sander. Dictionary of protein secondary structure : pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12) :2577–637, 1983.

[129] R. Karaduman, P. Fabrizio, K. Hartmuth, H. Urlaub, and R. Lührmann. RNA structure and RNA-protein interactions in purified yeast u6 snRNPs. *J Mol Biol*, 356(5) :1248–1262, Mar 2006.

[130] S. Karlin, Z. Y. Zhu, and F. Baud. Atom density in protein structures. *Proc Natl Acad Sci U S A*, 96(22) :12500–5, 1999.

[131] S. Karlin, M. Zuker, and L. Brocchieri. Measuring residue associations in protein structures. Possible implications fro protein folding. *J. Mol. Biol.*, 264 :121–136, 1994.

[132] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and I. A. Vakser. Molecular surface recognition : determination of geometric fit between proteins and their ligands by correlation techniques. *Proc Natl Acad Sci U S A*, 89(6) :2195–9, 1992.

[133] A. Ke and J. A. Doudna. Crystallization of RNA and RNA-protein complexes. *Methods*, 34(3) :408–14, 2004.

[134] J. C. Kendrew, R. E. Dickerson, B. E. Standberg, R. G. Hart, D. R. Davies, and D. C. Phillips. Structure of myoglobin. A three dimensional Fourier synthesis at 2 Å resolution. *Nature*, 185 :422–427, 1960.

[135] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3) :226–239, 1998.

[136] N. Kobayashi, T. Yamato, and N. Go. Mechanical property of a TIM-barrel protein. *Proteins*, 28(1) :109–16, 1997.

[137] K. Komatsu, Y. Kurihara, M. Iwadate, M. Takeda-Shitaka, and H. Umeyama. Evaluation of the third solvent clusters fitting procedure for the prediction of protein-protein interactions based on the results at the CAPRI blind docking study. *Proteins*, 52(1) :15–8, 2003.

[138] J. Konig, K. Zarnack, N. M. Luscombe, and J. Ule. Protein-RNA interactions : new genomic technologies and perspectives. *Nat Rev Genet*, 13(2) :77–83, 2011.

[139] O. Kotik-Kogan, E. R. Valentine, D. Sanfelice, M. R. Conte, and S. Curry. Structural analysis reveals conformational plasticity in the recognition of RNA 3' ends by the human La protein. *Structure*, 16(6) :852–862, Jun 2008.

[140] B. Krishnamoorthy and A. Tropsha. Development of a four-body statistical pseudo-potential to discriminate native from non-native protein conformations. *Bioinformatics*, 19(12) :1540–8, 2003.

[141] A. Kryshtafovych, C. Venclovas, K. Fidelis, and J. Moult. Progress over the first decade of CASP experiments. *Proteins*, 61 Suppl 7 :225–36, 2005.

[142] C. Laing and T. Schlick. Computational approaches to 3D modeling of RNA. *J Phys Condens Matter*, 22(28) :283101, 2010.

[143] D. S. Law, L. F. Ten Eyck, O. Katzenelson, I. Tsigelny, V. A. Roberts, M. E. Pique, and J. C. Mitchell. Finding needles in haystacks : Reranking DOT results by using shape complementarity, cluster analysis, and biological information. *Proteins*, 52(1) :33–40, 2003.

[144] T. Lazaridis and M. Karplus. Effective energy function for proteins in solution. *Proteins*, 35(2) :133–152, May 1999.

[145] R. H. Lee and G. D. Rose. Molecular recognition. I. Automatic identification of topographic surface features. *Biopolymers*, 24(8) :1613–27, 1985.

[146] M. F. Lensink, I. H. Moal, P. A. Bates, P. L. Kastritis, A. S. Melquiond, E. Karaca, C. Schmitz, M. van Dijk, A. M. Bonvin, M. Eisenstein, B. Jimenez-Garcia, S. Grosdidier, A. Solernou, L. Pérez-Cano, C. Pallara, J. Fernández-Recio, J. Xu, P. Muthu, K. Praneeth Kilambi, J. J. Gray, S. Grudinin, G. Derevyanko, J. C. Mitchell, J. Wieting, E. Kanamori, Y. Tsuchiya, Y. Murakami, J. Sarmiento, D. M. Standley, M. Shirota, K. Kinoshita, H. Nakamura, M. Chavent, D. W. Ritchie, H. Park, J. Ko, H. Lee, C. Seok, Y. Shen, D. Kozakov, S. Vajda, P. J. Kundrotas, I. A. Vakser, B. G. Pierce, H. Hwang, T. Vreven, Z. Weng, I. Buch, E. Farkash, H. J. Wolfson, M. Zacharias, S. Qin, H. X. Zhou, S. Y. Huang, X. Zou, J. A. Wojdyla, C. Kleanthous, and S. J. Wodak. Blind prediction of interfacial water positions in CAPRI. *Proteins*, 2013.

[147] M. F. Lensink, R. Méndez, and S. J. Wodak. Docking and scoring protein complexes : CAPRI 3rd edition. *Proteins*, 69(4) :704–718, Dec 2007.

[148] M. F. Lensink and S. J. Wodak. Docking, scoring, and affinity prediction in CAPRI. *Proteins*, 81(12) :2082–95, 2013.

[149] C. Levinthal, S. J. Wodak, P. Kahn, and A. K. Dadivanian. Hemoglobin interaction in sickle cell fibers. I : theoretical approaches to the molecular contacts. *Proc Natl Acad Sci U S A*, 72(4) :1330–4, 1975.

[150] M. Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *J Mol Biol*, 104(1) :59–107, 1976.

[151] M. Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *J Mol Biol*, 104(1) :59–107, Jun 1976.

[152] B. A. Lewis, R. R. Walia, M. Terribilini, J. Ferguson, C. Zheng, V. Honavar, and D. Dobbs. PRIDB : a protein-RNA interface database. *Nucleic Acids Res*, 39(Database issue) :D277–D282, Jan 2011.

[153] C. H. Li, L. B. Cao, J. G. Su, Y. X. Yang, and C. X. Wang. A new residue-nucleotide propensity potential with structural information considered for discriminating protein-RNA docking decoys. *Proteins*, 80(1) :14–24, 2012.

[154] L. Li, R. Chen, and Z. Weng. RDOCK : refinement of rigid-body protein docking predictions. *Proteins*, 53(3) :693–707, 2003.

[155] W. Li, H. Yoshii, N. Hori, T. Kameda, and S. Takada. Multiscale methods for protein folding simulations. *Methods*, 52(1) :106–114, Sep 2010.

[156] X. Li, C. Hu, and J. Liang. Simplicial edge representation of protein structures and alpha contact potential with confidence measure. *Proteins*, 53(4) :792–805, 2003.

[157] J. Liang and K. A. Dill. Are proteins well-packed ? *Biophys J*, 81(2) :751–66, 2001.

[158] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules : I. Molecular area and volume through alpha shape. *Proteins*, 33(1) :1–17, 1998.

[159] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules : II. Inaccessible cavities in proteins. *Proteins*, 33(1) :18–29, 1998.

[160] J. Liang, H. Edelsbrunner, and C. Woodward. Anatomy of protein pockets and cavities : measurement of binding site geometry and implications for ligand design. *Protein Sci*, 7(9) :1884–97, 1998.

[161] J. Liang and S. Subramaniam. Computation of molecular electrostatics with boundary element methods. *Biophys J*, 73(4) :1830–41, 1997.

[162] D. R. Lide. *CRC handbook of chemistry and physics*. CRC press, 2004.

[163] S. L. Lin, R. Nussinov, D. Fischer, and H. J. Wolfson. Molecular surface representations by sparse critical points. *Proteins*, 18(1) :94–101, 1994.

[164] C. X. Ling, J. Huang, and H. Zhang. AUC : a better measure than accuracy in comparing learning algorithms. *Advances in Artificial Intelligence*, Jan 2003.

[165] C. X. Ling, J. Huang, and H. Zhang. AUC : a statistically consistent and more discriminating measure than accuracy. *International joint Conference on artificial intelligence*, pages 519–524, Jan 2003.

[166] J. Lipfert and S. Doniach. Small-angle X-ray scattering from RNA, proteins, and protein complexes. *Annu Rev Biophys Biomol Struct*, 36 :307–27, 2007.

[167] Y. Liu and J. Snoeyink. A comparison of five implementations of 3D Delaunay tessellation. *Combinatorial and Computational Geometry*, 52 :439–458, 2005.

[168] S. Loriot, F. Cazals, and J. Bernauer. ESBTL : efficient PDB parser and data structure for the structural and geometric analysis of biological macromolecules. *Bioinformatics*, 26(8) :1127–1128, Apr 2010.

[169] D. Luo, T. Xu, R. P. Watson, D. Scherer-Becker, A. Sampath, W. Jahnke, S. S. Yeong, C. H. Wang, S. P. Lim, A. Strongin, et al. Insights into rna unwinding and atp hydrolysis by the flavivirus ns3 protein. *The EMBO journal*, 27(23) :3209–3219, 2008.

[170] J. G. Mandell, V. A. Roberts, M. E. Pique, V. Kotlovyi, J. C. Mitchell, E. Nelson, I. Tsigelny, and L. F. Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Eng*, 14(2) :105–13, 2001.

[171] D. H. Mathews. Revolutions in RNA secondary structure prediction. *J Mol Biol*, 359(3) :526–532, Jun 2006.

[172] B. J. McConkey, V. Sobolev, and M. Edelman. Quantification of protein surfaces, volumes and atom-atom contacts using a constrained Voronoi procedure. *Bioinformatics*, 18(10) :1365–73, 2002.

[173] R. Mendez, R. Leplae, L. De Maria, and S. J. Wodak. Assessment of blind predictions of protein-protein interactions : current status of docking methods. *Proteins*, 52(1) :51–67, 2003.

[174] R. Mendez, R. Leplae, M. F. Lensink, and S. J. Wodak. Assessment of CAPRI predictions in rounds 3-5 shows progress in docking procedures. *Proteins*, 60(2) :150–69, 2005.

[175] E. C. Meng, B. K. Shoichet, and I. D. Kuntz. Automated docking with grid-based energy evaluation. *J Comp Chem*, 13 :505–524, 1992.

[176] E. J. Merino, K. A. Wilkinson, J. L. Coughlan, and K. M. Weeks. RNA structure analysis at single nucleotide resolution by selective 2'-hydroxyl acylation and primer extension (shape). *J Am Chem Soc*, 127(12) :4223–4231, Mar 2005.

[177] C. E. Metz. Basic principles of ROC analysis. *Seminars in nuclear medicine*, VIII(4) :283–298, Jan 1978.

[178] I. Mihalek, I. Res, and O. Lichtarge. A structure and evolution-guided Monte Carlo sequence selection strategy for multiple alignment-based analysis of proteins. *Bioinformatics*, 22(2) :149–56, 2006.

[179] M. Milek, E. Wyler, and M. Landthaler. Transcriptome-wide analysis of protein-RNA interactions using high-throughput sequencing. *Semin Cell Dev Biol*, 23(2) :206–12, 2012.

[180] J. C. Mitchell, R. Kerr, and L. F. Ten Eyck. Rapid atomic density methods for molecular shape characterization. *J Mol Graph Model*, 19(3-4) :325–30, 388–90, 2001.

[181] I. S. Moreira, P. A. Fernandes, and M. J. Ramos. Protein-protein docking dealing with the unknown. *J Comput Chem*, 31(2) :317–42, 2010.

[182] J. Moult, K. Fidelis, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP)–round 6. *Proteins*, 61 Suppl 7 :3–7, 2005.

[183] P. J. Munson and R. K. Singh. Statistical significance of hierarchical multi-body potentials based on Delaunay tessellation and their application in sequence-structure alignment. *Protein Sci*, 6(7) :1467–81, 1997.

[184] G. N. Murshudov, A. A. Vagin, and E. J. Dodson. Refinement of macromolecular structures by the maximum-likelihood method. *Acta Crystallographica Section D : Biological Crystallography*, 53(3) :240–255, 1997.

[185] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP : a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4) :536–40, 1995.

[186] K. Nadassy, I. Tomas-Oliveira, I. Alberts, J. Janin, and S. J. Wodak. Standard atomic volumes in double-stranded DNA and packing in protein–DNA interfaces. *Nucleic Acids Res*, 29(16) :3362–76, 2001.

[187] K. Nadassy, S. J. Wodak, and J. Janin. Structural features of protein-nucleic acid recognition sites. *Biochemistry*, 38(7) :1999–2017, 1999.

[188] R. Norel, D. Fischer, H. J. Wolfson, and R. Nussinov. Molecular surface recognition by a computer vision-based technique. *Protein Eng*, 7(1) :39–46, 1994.

[189] R. Norel, S. L. Lin, H. J. Wolfson, and R. Nussinov. Molecular surface complementarity at protein-protein interfaces : the critical role played by surface normals at well placed, sparse, points in docking. *J Mol Biol*, 252(2) :263–73, 1995.

[190] R. Norel, D. Petrey, H. J. Wolfson, and R. Nussinov. Examination of shape complementarity in docking of unbound proteins. *Proteins*, 36(3) :307–17, 1999.

[191] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8) :1093–108, 1997.

[192] E. Paci and M. Marchi. Intrinsic compressibility and volume compression in solvated proteins by molecular dynamics simulation at high pressure. *Proc Natl Acad Sci U S A*, 93(21) :11609–14, 1996.

[193] H. Pan, S. Agarwalla, D. T. Moustakas, J. Finer-Moore, and R. M. Stroud. Structure of tRNA pseudouridine synthase TruB and its RNA complex : RNA recognition through a combination of rigid docking and induced fit. *Proc Natl Acad Sci U S A*, 100(22) :12648–12653, Oct 2003.

[194] P. Pancoska and T. A. Keiderling. Systematic comparison of statistical analyses of electronic and vibrational circular dichroism for secondary structure prediction of selected proteins. *Biochemistry*, 30(28) :6885–95, 1991.

[195] M. Parisien and F. Major. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183) :51–5, 2008.

[196] L. Pauling and R. B. Corey. The pleated sheet, a new layer configuration of polypeptide chains. *Proc Natl Acad Sci U S A*, 37(5) :251–6, 1951.

[197] L. Pauling, R. B. Corey, and H. R. Branson. The structure of proteins ; two hydrogen-bonded helical configurations of the polypeptide chain. *Proc Natl Acad Sci U S A*, 37(4) :205–11, 1951.

[198] K. P. Peters, J. Fauck, and C. Frommel. The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. *J Mol Biol*, 256(1) :201–13, 1996.

[199] B. Pierce, W. Tong, and Z. Weng. M-ZDOCK : a grid-based approach for Cn symmetric multimer docking. *Bioinformatics*, 21(8) :1472–8, 2005.

[200] H. Ponstingl, K. Henrick, and J. M. Thornton. Discriminating between homodimeric and monomeric proteins in the crystalline state. *Proteins*, 41(1) :47–57, 2000.

[201] A. Poupon. Voronoi and Voronoi-related tessellations in studies of protein structure and interaction. *Curr Opin Struct Biol*, 14(2) :233–41, 2004.

[202] R. Pribic, I. H. van Stokkum, D. Chapman, P. I. Haris, and M. Bloemendal. Protein secondary structure from Fourier transform infrared and/or circular dichroism spectra. *Anal Biochem*, 214(2) :366–78, 1993.

[203] T. Puton, L. Kozlowski, I. Tuszynska, K. Rother, and J. M. Bujnicki. Computational methods for prediction of protein-RNA interactions. *J Struct Biol*, 179(3) :261–8, 2012.

[204] L. Pérez-Cano, B. Jiménez-García, and J. Fernández-Recio. A protein-RNA docking benchmark (II) : extended set from experimental and homology modeling data. *Proteins*, 80(7) :1872–1882, Jul 2012.

[205] L. Pérez-Cano, A. Solernou, C. Pons, and J. Fernández-Recio. Structural prediction of protein-RNA interaction by computational docking with propensity-based statistical potentials. *Pac Symp Biocomput*, pages 293–301, 2010.

[206] M. L. Quillin and B. W. Matthews. Accurate calculation of the density of proteins. *Acta Crystallogr D Biol Crystallogr*, 56 (Pt 7) :791–4, 2000.

[207] J. R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[208] A. Rakotomamonjy. Optimizing area under ROC curves with SVMs. *ROCAI'04*, Jan 2004.

[209] G. N. Ramachandran. Conformation of polypeptides and proteins. *Advances in protein chemistry*, 23 :283, 1968.

[210] J. Reeder, M. Höchsmann, M. Rehmsmeier, B. Voss, and R. Giegerich. Beyond Mfold : recent advances in RNA bioinformatics. *J Biotechnol*, 124(1) :41–55, Jun 2006.

[211] F. M. Richards. The interpretation of protein structures : total volume, group volume distributions and packing density. *J Mol Biol*, 82(1) :1–14, 1974.

[212] F. M. Richards. Calculation of molecular volumes and areas for structures of known geometry. *Methods Enzymol*, 115 :440–64, 1985.

[213] D. W. Ritchie. Evaluation of protein docking predictions using Hex 3.1 in CAPRI rounds 1 and 2. *Proteins*, 52(1) :98–106, 2003.

[214] M. Roche, J. Azé, Y. Kodratoff, and M. Sebag. Learning interestingness measures in terminology extraction. a ROC-based approach. In José Hernández-Orallo, César Ferri, Nicolas Lachiche, and Peter A. Flach, editors, *ROC Analysis in Artificial Intelligence, 1st International Workshop, ROCAI-2004, Valencia, Spain, August 22, 2004*, ROCAI, pages 81–88, 2004.

[215] G. D. Rose, A. R. Geselowitz, G. J. Lesser, R. H. Lee, and M. H. Zehfus. Hydrophobicity of amino acid residues in globular proteins. *Science*, 229(4716) :834–8, 1985.

[216] B. Rost, C. Sander, and R. Schneider. Redefining the goals of protein secondary structure prediction. *J Mol Biol*, 235(1) :13–26, 1994.

[217] K. Rother, M. Rother, M. Boniecki, T. Puton, and J. M. Bujnicki. RNA and protein 3D structure modeling : similarities and differences. *J Mol Model*, 17(9) :2325–36, 2011.

[218] M. Ruff, S. Krishnaswamy, M. Boeglin, A. Poterszman, A. Mitschler, A. Podjarny, B. Rees, J. C. Thierry, and D. Moras. Class II aminoacyl transfer RNA synthetases : crystal structure of yeast aspartyl-tRNA synthetase complexed with tRNA (asp). *Science*, 252(5013) :1682–1689, 1991.

[219] S. P. Ryder and S. A. Strobel. Nucleotide analog interference mapping. *Methods*, 18(1) :38–50, May 1999.

[220] B. Sandak, R. Nussinov, and H. J. Wolfson. A method for biomolecular structural recognition and docking allowing conformational flexibility. *J Comput Biol*, 5(4) :631–54, 1998.

[221] C. Sander and R. Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9(1) :56–68, 1991.

[222] M. Schaefer, C. Bartels, F. Leclerc, and M. Karplus. Effective atom volumes for implicit solvent models : comparison between Voronoi volumes and minimum fluctuation volumes. *J Comput Chem*, 22(15) :1857–1879, 2001.

[223] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin : a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5) :1651–1686, 10 1998.

[224] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson. Geometry-based flexible and symmetric protein docking. *Proteins*, 60(2) :224–31, 2005.

[225] D. Schneidman-Duhovny, Y. Inbar, V. Polak, M. Shatsky, I. Halperin, H. Benyamini, A. Barzilai, O. Dror, N. Haspel, R. Nussinov, and H.J. Wolfson. Taking geometry to its edge : fast unbound rigid (and hinge-bent) docking. *Proteins*, 52(1) :107–12, 2003.

[226] O. Schueler-Furman, C. Wang, and D. Baker. Progress in protein-protein docking : atomic resolution predictions in the CAPRI experiment using RosettaDock with an improved treatment of side-chain flexibility. *Proteins*, 60(2) :187–94, 2005.

[227] L. G. Scott and M. Hennig. RNA structure determination by NMR. *Methods Mol Biol*, 452 :29–61, 2008.

[228] M. Sebag, J. Azé, and N. Lucas. Impact studies and sensitivity analysis in medical data mining with ROC-based genetic learning. In *Proceedings of the Third IEEE International Conference on Data Mining, ICDM 2003*, pages 637–40, Nov 2003.

[229] P. Setny and M. Zacharias. A coarse-grained force field for protein-RNA docking. *Nucleic Acids Res*, 39(21) :9118–29, 2011.

[230] B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald. Bridging the gap in RNA structure prediction. *Curr Opin Struct Biol*, 17(2) :157–165, Apr 2007.

[231] Paul Sherwood, Bernard R. Brooks, and Mark S P. Sansom. Multiscale methods for macromolecular simulations. *Curr Opin Struct Biol*, 18(5) :630–640, Oct 2008.

[232] B. K. Shoichet, I. D. Kuntz, and D. L. Bodian. Molecular docking using shape descriptors. *J Comp Chem*, 13 :380–397, 1992.

[233] R. K. Singh, A. Tropsha, and I. I. Vaisman. Delaunay tessellation of proteins : four body nearest-neighbor propensities of amino acid residues. *J Comput Biol*, 3(2) :213–21, 1996.

[234] G. R. Smith and M. J. Sternberg. Prediction of protein-protein interactions by docking methods. *Curr Opin Struct Biol*, 12(1) :28–35, 2002.

[235] G. R. Smith and M. J. Sternberg. Evaluation of the 3D-Dock protein docking suite in rounds 1 and 2 of the CAPRI blind trial. *Proteins*, 52(1) :74–9, 2003.

[236] A. Soyer, J. Chomilier, J.P. Mornon, R. Jullien, and J.F. Sadoc. Voronoi tessellation reveals the condensed matter character of folded proteins. *Phys Rev Lett*, 85(16) :3532–5, 2000.

[237] M. L. Stolowitz. Chemical protein sequencing and amino acid analysis. *Curr Opin Biotechnol*, 4(1) :9–13, 1993.

[238] G. Terashi, M. Takeda-Shitaka, D. Takaya, K. Komatsu, and H. Umeyama. Searching for protein-protein interaction sites and docking by the methods of molecular dynamics, grid scoring, and the pairwise interaction potential of amino acid residues. *Proteins*, 60(2) :289–95, 2005.

[239] C. A. Theimer, N. L. Smith, and M. Khanna. NMR studies of protein-RNA interactions. *Methods Mol Biol*, 831 :197–218, 2012.

[240] R. Thiele, R. Zimmer, and T. Lengauer. Protein threading by recursive dynamic programming. *J Mol Biol*, 290(3) :757–79, 1999.

[241] P. Tijerina, S. Mohr, and R. Russell. DMS footprinting of structured RNAs and RNA-protein complexes. *Nat Protoc*, 2(10) :2608–2623, 2007.

[242] V. Tozzini. Coarse-grained models for proteins. *Curr Opin Struct Biol*, 15(2) :144–150, Apr 2005.

[243] J. Tsai and M. Gerstein. Calculations of protein volumes : sensitivity analysis and parameter database. *Bioinformatics*, 18(7) :985–95, 2002.

[244] J. Tsai, R. Taylor, C. Chothia, and M. Gerstein. The packing density in proteins : standard radii and volumes. *J Mol Biol*, 290(1) :253–66, 1999.

[245] J. Tsai, N. Voss, and M. Gerstein. Determining the minimum number of types necessary to represent the sizes of protein atoms. *Bioinformatics*, 17(10) :949–56, 2001.

[246] T. D. Tullius and B. A. Dombroski. Hydroxyl radical "footprinting" : high-resolution information about DNA-protein contacts and application to lambda repressor and Cro protein. *Proc Natl Acad Sci U S A*, 83(15) :5469–5473, Aug 1986.

[247] I. Tuszynska and J. M. Bujnicki. DARS-RNP and QUASI-RNP : new statistical potentials for protein-RNA docking. *BMC Bioinformatics*, 12 :348, 2011.

[248] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770) :623–7, 2000.

[249] I. A. Vakser and P. Kundrotas. Predicting 3D structures of protein-protein complexes. *Curr Pharm Biotechnol*, 9(2) :57–66, 2008.

[250] A. D. van Dijk, S. J. de Vries, C. Dominguez, H. Chen, H. X. Zhou, and A. M. Bonvin. Data-driven docking : HADDOCK's adventures in CAPRI. *Proteins*, 60(2) :232–8, 2005.

[251] S. Viswanath, D. V. Ravikant, and R. Elber. Improving ranking of models for protein complexes with side chain modeling and atomic potentials. *Proteins*, 81(4) :592–606, 2013.

[252] M. Vuk and T. Curk. ROC curve, lift chart and calibration plot. *Metodoloski zvezki*, 3(1) :89–108, Jan 2006.

[253] H. Wako and T. Yamato. Novel method to detect a motif of local structures in different protein conformations. *Protein Eng*, 11(11) :981–90, 1998.

[254] R. R. Walia, C. Caragea, B. A. Lewis, F. Towfic, M. Terribilini, Y. El-Manzalawy, D. Dobbs, and V. Honavar. Protein-RNA interface residue prediction using machine learning : an assessment of the state of the art. *BMC Bioinformatics*, 13 :89, 2012.

[255] H. Wang. Grid-search molecular accessible surface algorithm for solving the protein docking problem. *J Comp Chem*, 12 :746–750, 1991.

[256] L. Wernisch, M. Hunting, and S. J. Wodak. Identification of structural domains in proteins by a graph heuristic. *Proteins*, 35(3) :338–52, 1999.

[257] K. Wiehe, B. Pierce, J. Mintseris, W. W. Tong, R. Anderson, R. Chen, and Z. Weng. ZDOCK and RDOCK performance in CAPRI rounds 3, 4, and 5. *Proteins*, 60(2) :207–13, 2005.

[258] D. S. Wishart, B. D. Sykes, and F. M. Richards. Relationship between nuclear magnetic resonance chemical shift and protein secondary structure. *J Mol Biol*, 222(2) :311–33, 1991.

[259] S. J. Wodak and J. Janin. Computer analysis of protein-protein interaction. *J Mol Biol*, 124(2) :323–42, 1978.

[260] S. J. Wodak and J. Janin. Structural basis of macromolecular recognition. *Adv Protein Chem*, 61 :9–73, 2002.

[261] S. J. Wodak and R. Mendez. Prediction of protein-protein interactions : the CAPRI experiment, its evaluation and implications. *Curr Opin Struct Biol*, 14(2) :242–9, 2004.

[262] H. J. Wolfson and R. Nussinov. Geometrical docking algorithms. A practical approach. *Methods Mol Biol*, 143 :377–97, 2000.

[263] X. Yang, T. Gérczei, L. Glover, and C. C. Correll. Crystal structures of restrictocin–inhibitor complexes with implications for RNA recognition and base flipping. *Nature Structural & Molecular Biology*, 8(11) :968–973, 2001.

[264] K.-D. Zachmann, W. Heiden, M. Schlenkrich, and J. Brickmann. Topological analysis of complex molecular surfaces. *J Comp Chem*, 13 :76–84, 1992.

[265] C. Zhang, S. Liu, and Y. Zhou. Accurate and efficient loop selections by the DFIRE-based all-atom statistical potential. *Protein Sci*, 13(2) :391–9, 2004.

[266] C. Zhang, S. Liu, and Y. Zhou. Docking prediction using biological information, ZDOCK sampling technique, and clustering guided by the DFIRE statistical energy function. *Proteins*, 60(2) :314–8, 2005.

[267] C. Zhang, S. Liu, Q. Zhu, and Y. Zhou. A knowledge-based energy function for protein-ligand, protein-protein, and protein-DNA complexes. *J Med Chem*, 48(7) :2325–35, 2005.

[268] J. Zhang. Protein-length distributions for the three domains of life. *Trends Genet*, 16(3) :107–9, 2000.

[269] S. Zheng, T. A. Robertson, and G. Varani. A knowledge-based potential function predicts the specificity and relative binding energy of RNA-binding proteins. *FEBS J*, 274(24) :6378–91, 2007.

[270] Z. H. Zhou. Towards atomic resolution structural determination by single-particle cryo-electron microscopy. *Curr Opin Struct Biol*, 18(2) :218–28, 2008.

[271] H. Zhu, F. S. Domingues, I. Sommer, and T. Lengauer. NOXclass : prediction of protein-protein interaction types. *BMC Bioinformatics*, 7(1) :27, 2006.

[272] X. Zhu, S. S. Ericksen, O. N. Demerdash, and J. C. Mitchell. Data-driven models for protein interaction and design. *Proteins*, 81(12) :2221–8, 2013.

[273] R. Zimmer, M. Wohler, and R. Thiele. New scoring schemes for protein fold recognition based on Voronoi contacts. *Bioinformatics*, 14(3) :295–308, 1998.

[274] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res*, 31(13) :3406–3415, Jul 2003.

Annexes

```
1  # Perturbation.
2  -docking:dock_pert 3 8 # Perturb a little the second partner (3A, 8d).
3  # Prepacking.
4  -docking:dock_ppk false # No docking prepack mode.
5  # Output.
6  -out:pdb # Output pdb file.
7  -out:overwrite true # Overwrite output files.
8  # Docking options.
9  -docking:docking_centroid_outer_cycles 0
10 -docking:docking_centroid_inner_cycles 0
11 -docking:no_filters true
12 -docking:dock_mcm false
13 -docking:sc_min false
14 -docking:dock_min false
```

Listing .1 – Fichier de *flags* pour générer des candidats par perturbation.

```
1  # Perturbation.
2  -docking:dock_pert 3 8 # Perturb a little the second partner (3A, 8d).
3  -docking:randomize2 # Randomize the second partner (partner B).
4  -docking:spin # Spin a little the second partner.
5  # Prepacking.
6  -docking:dock_ppk  true # Docking prepack mode.
7  # Output.
8  -out:pdb # Output pdb file.
9  -out:overwrite
10 # Docking options.
11 -docking:docking_local_refine true
```

Listing .2 – Fichier de *flags* pour générer des candidats par amarrage atomique.

```
1  # Perturbation.
2  -docking:dock_pert 3 8 # Perturb a little the second partner (3A, 8d).
3  -docking:randomize2 # Randomize the second partner (partner B).
4  -docking:spin # Spin a little the second partner.
5  # Prepacking.
6  -docking:dock_ppk  true # Docking prepack mode.
7  # Output.
8  -out:pdb # Output pdb file.
9  -out:overwrite
10 # Docking options.
11 -docking:low_res_protocol_only # Skip high res docking.
```

Listing .3 – Fichier de *flags* pour générer des candidats par amarrage gros-grain.

```
1  # Perturbation.
2  -docking:dock_pert 3 8 # Perturb a little the second partner (3A, 8d).
3  -docking:randomize2 # Randomize the second partner (partner B).
4  -docking:spin # Spin a little the second partner.
5  # Prepacking.
6  -docking:dock_ppk  true # Docking prepack mode.
7  # Output.
8  -out:pdb # Output pdb file.
9  -out:overwrite true # Overwrite output files.
```

Listing .4 – Fichier de *flags* pour générer des candidats par amarrage en aveugle.

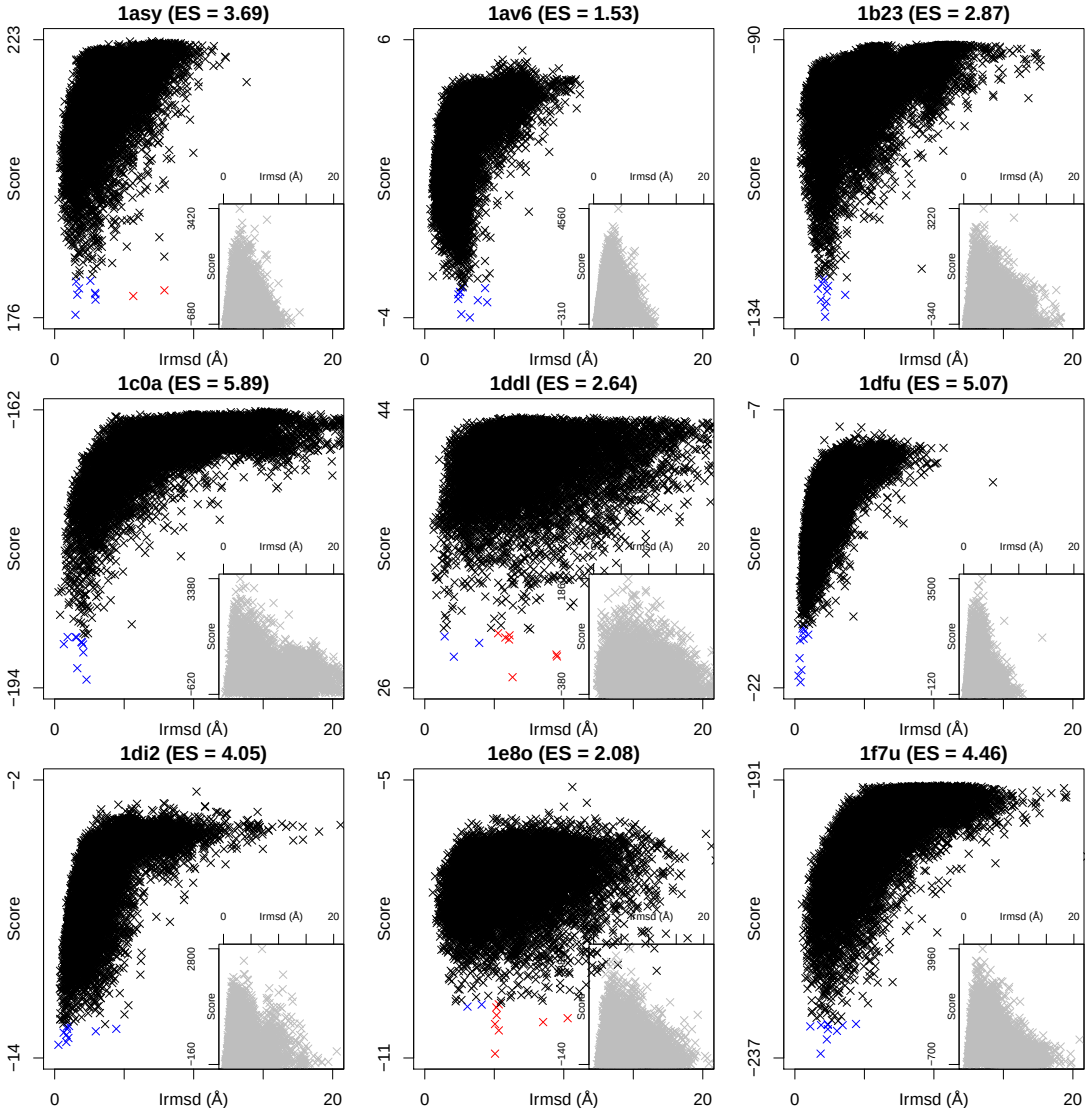


FIGURE S1 – Diagramme par complexe de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

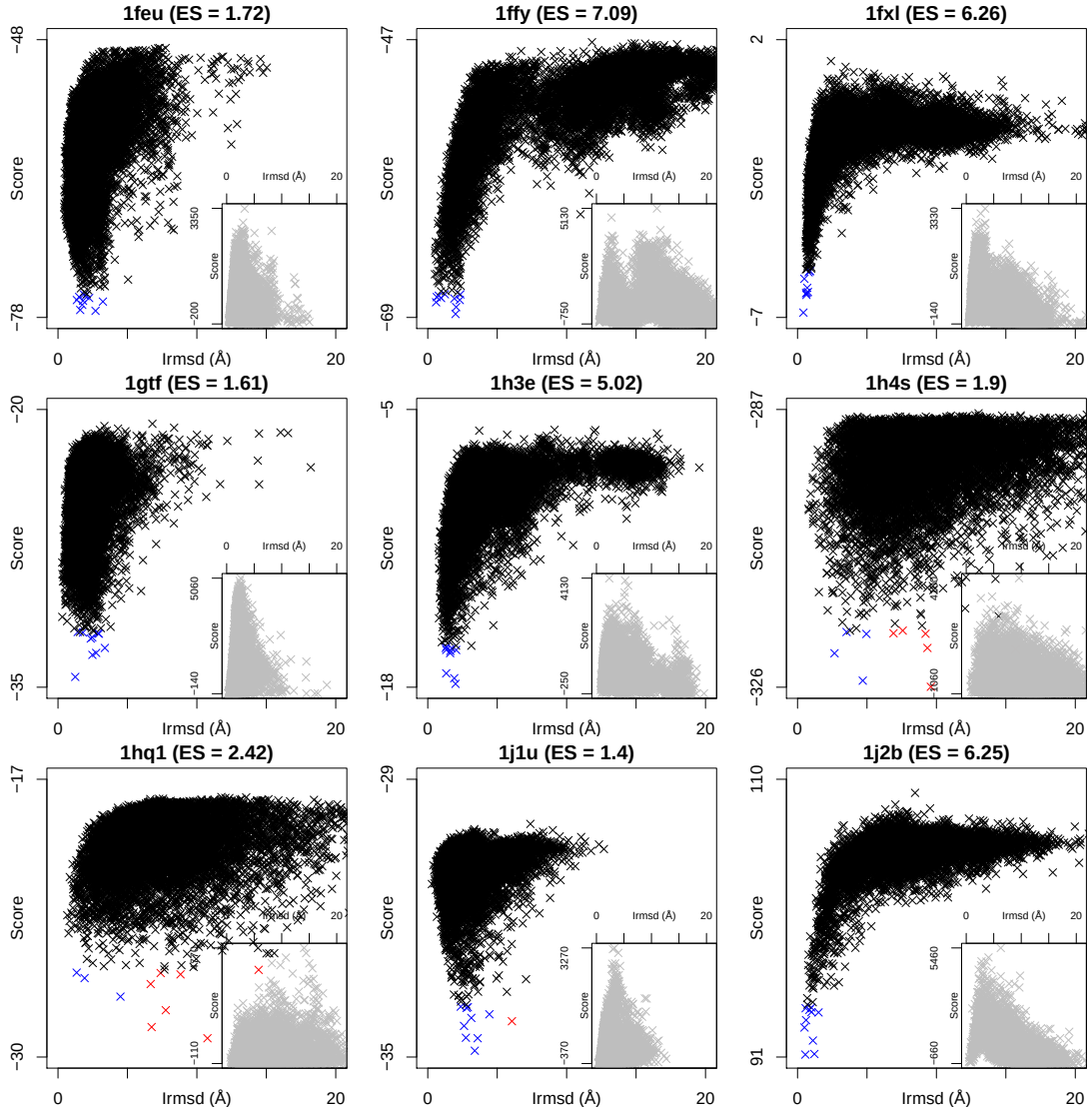


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

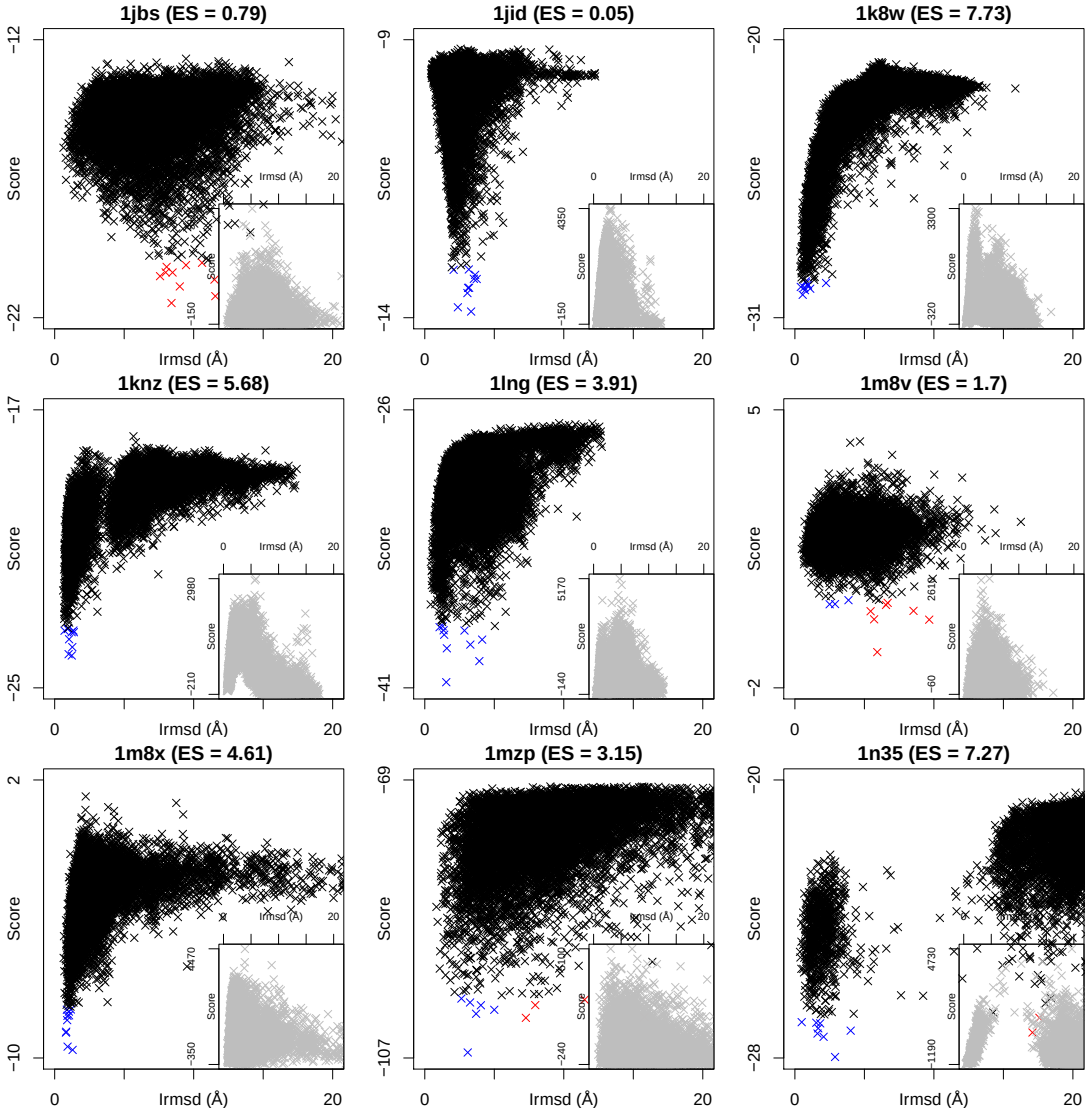


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

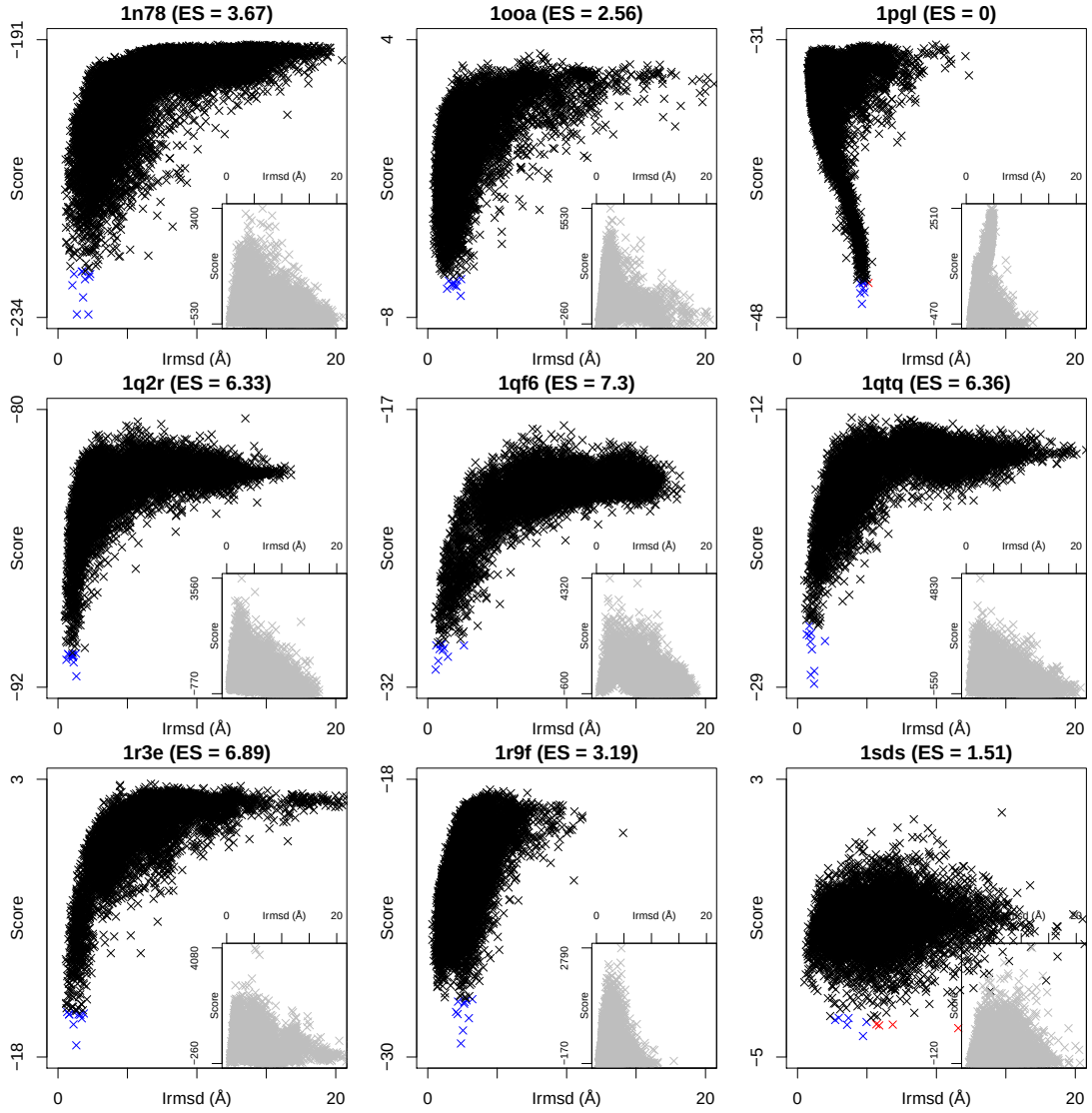


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

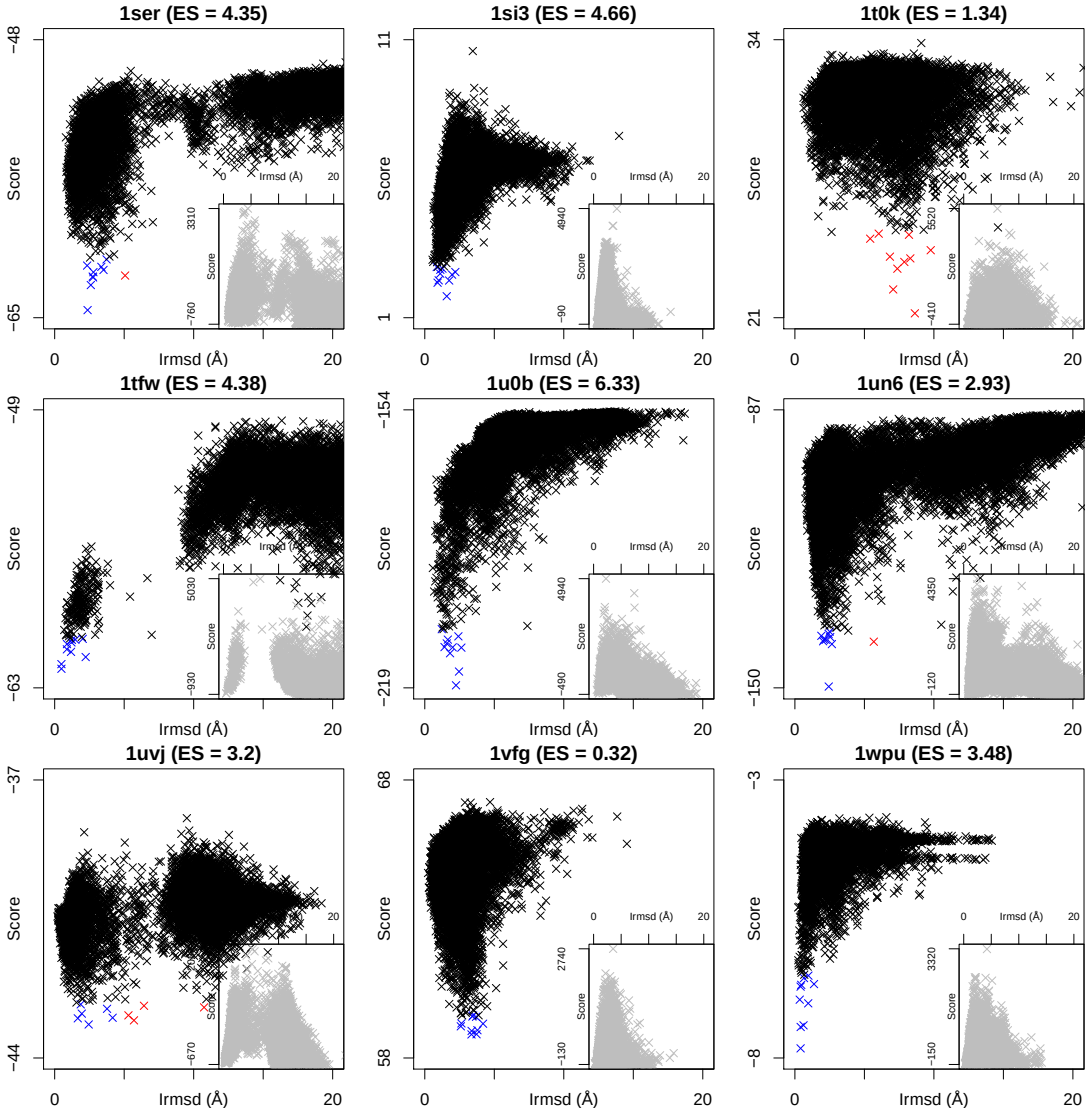


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

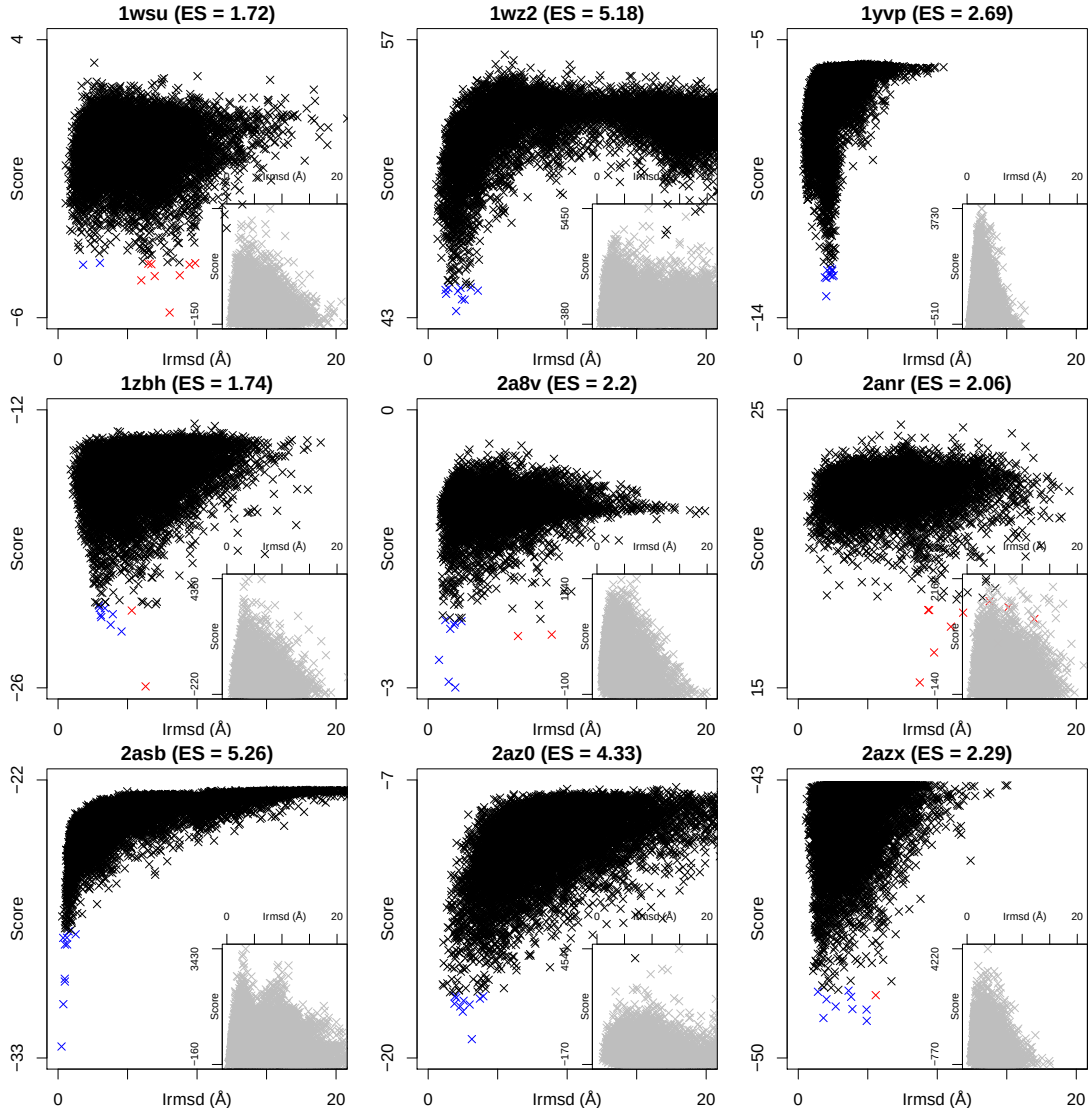


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

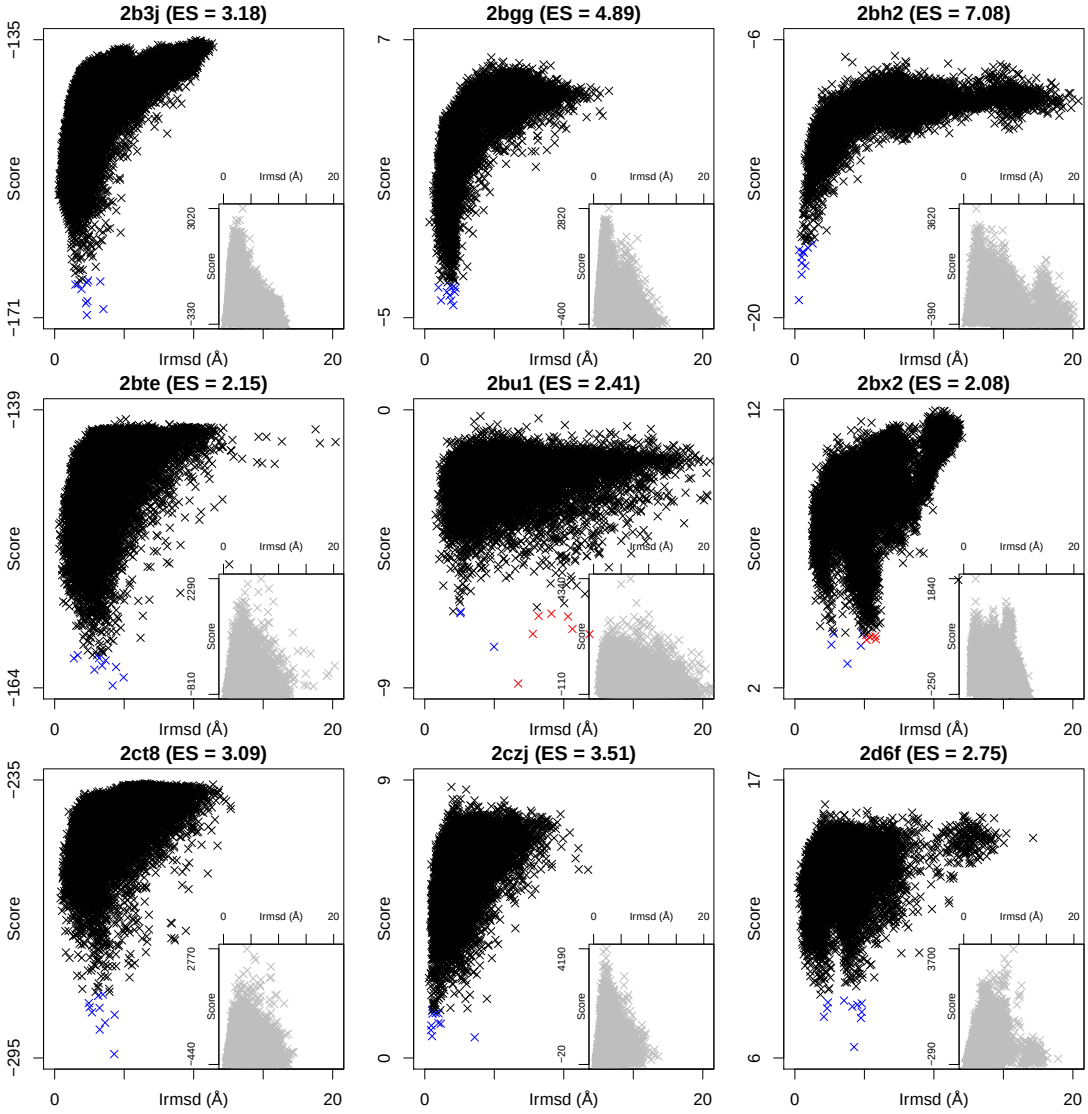


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

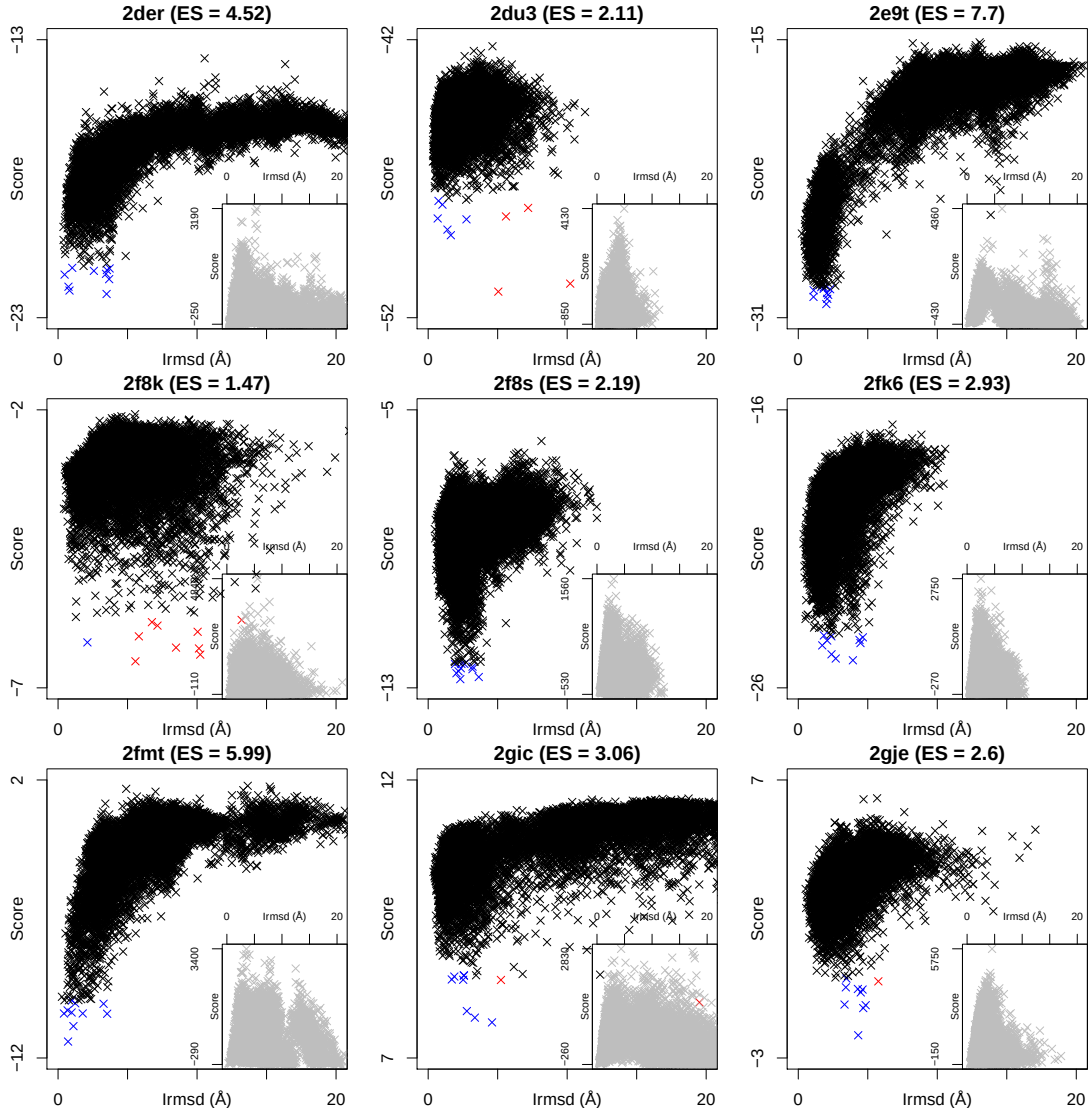


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

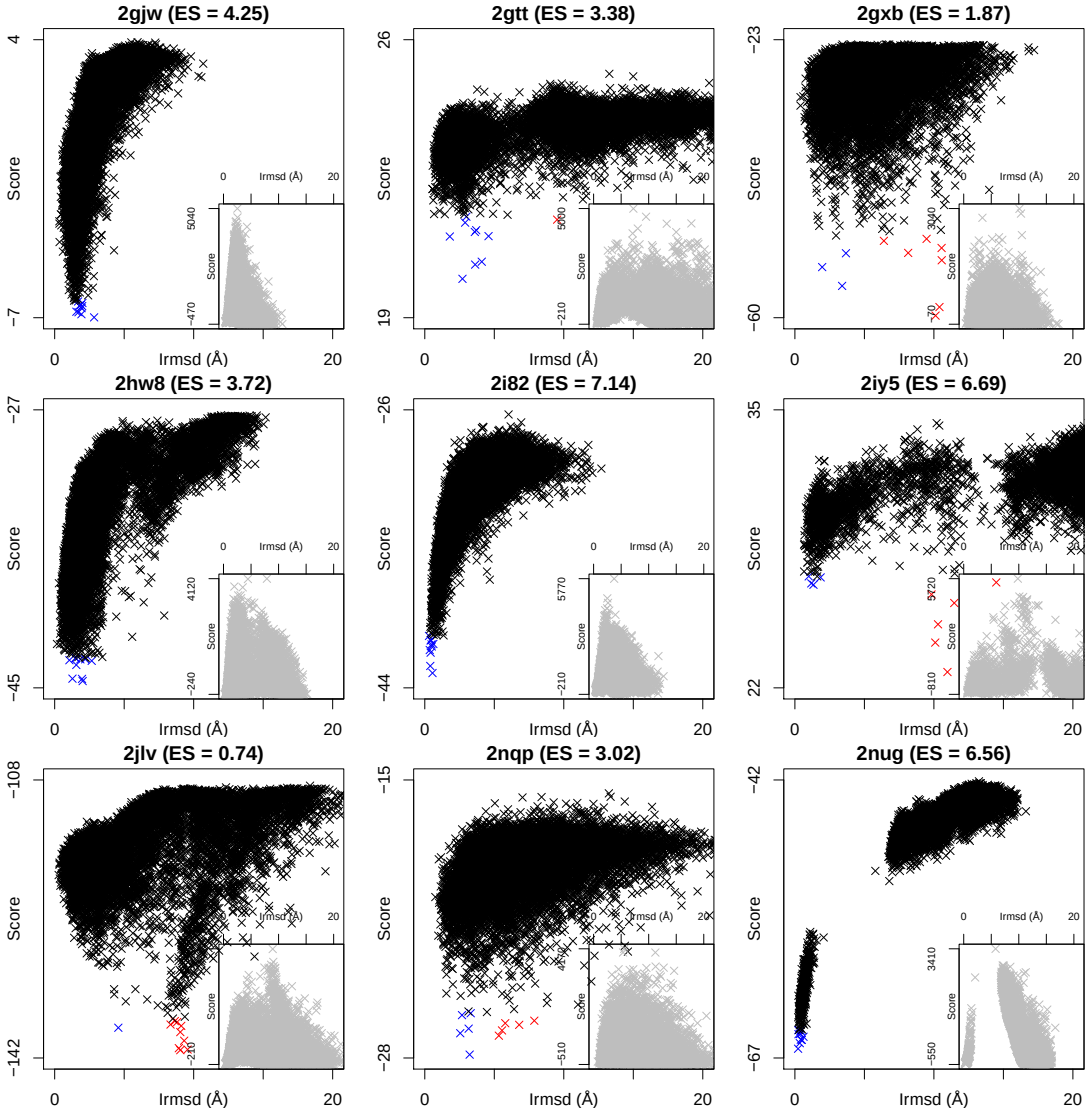


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

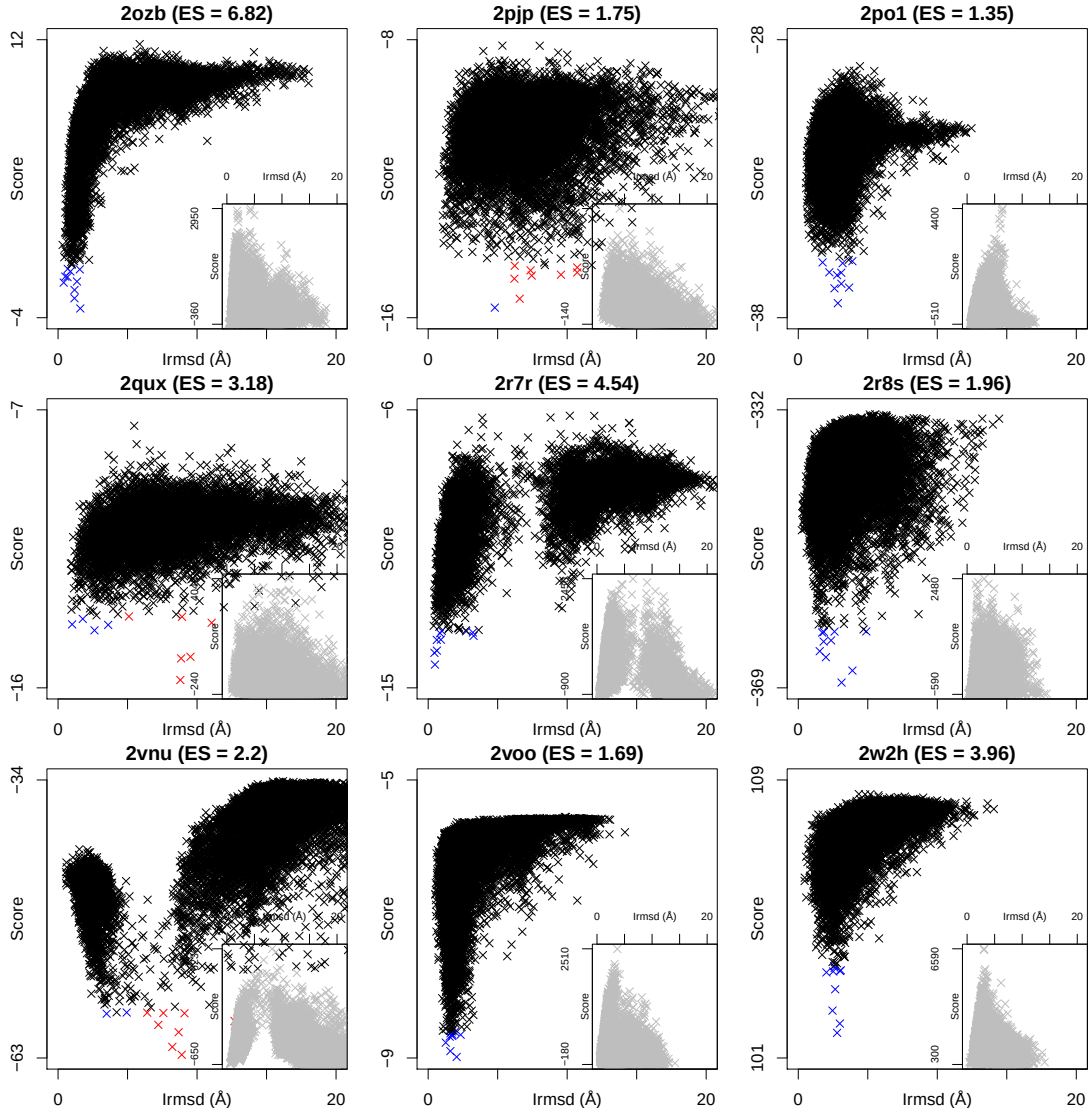


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

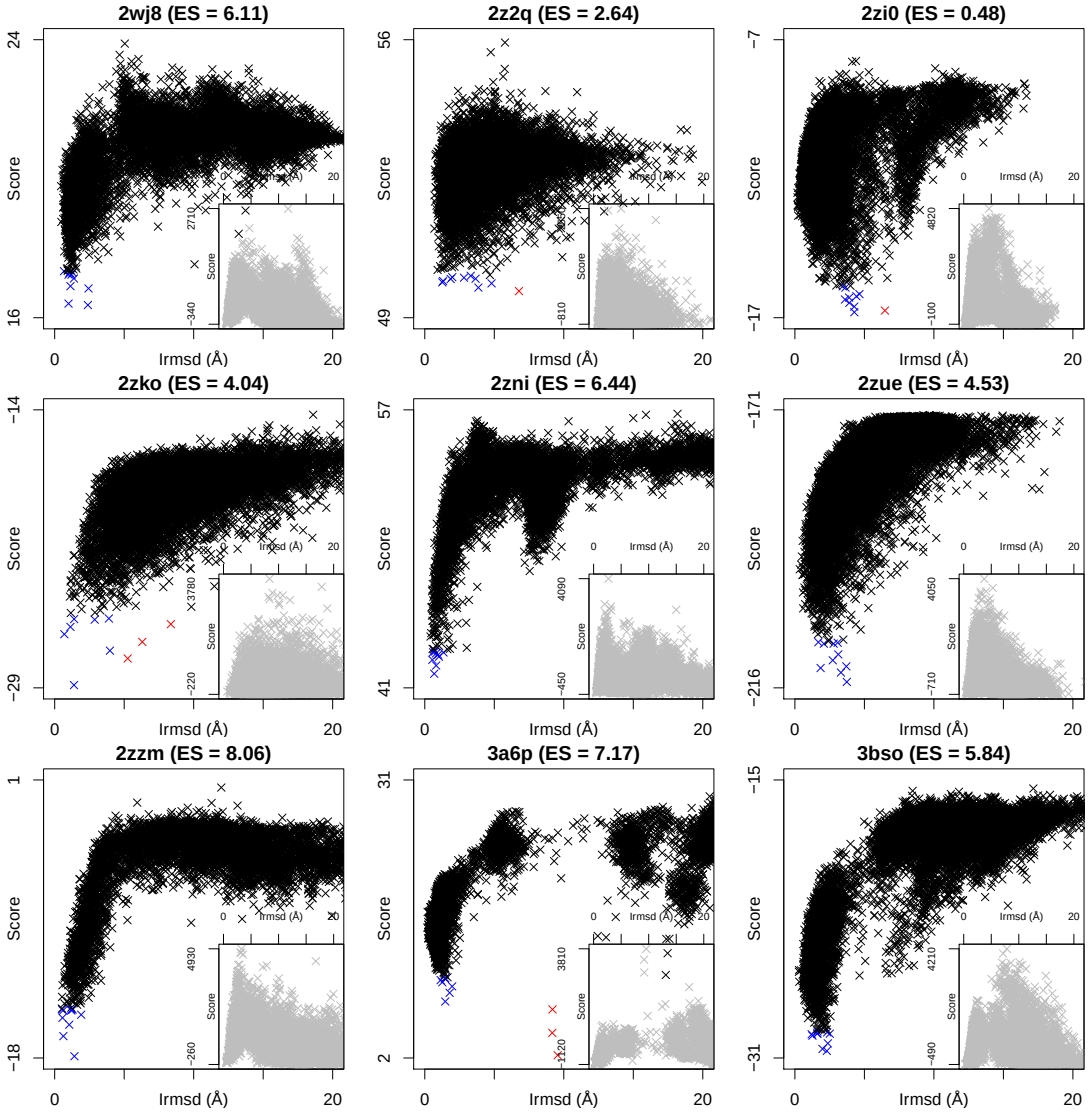


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

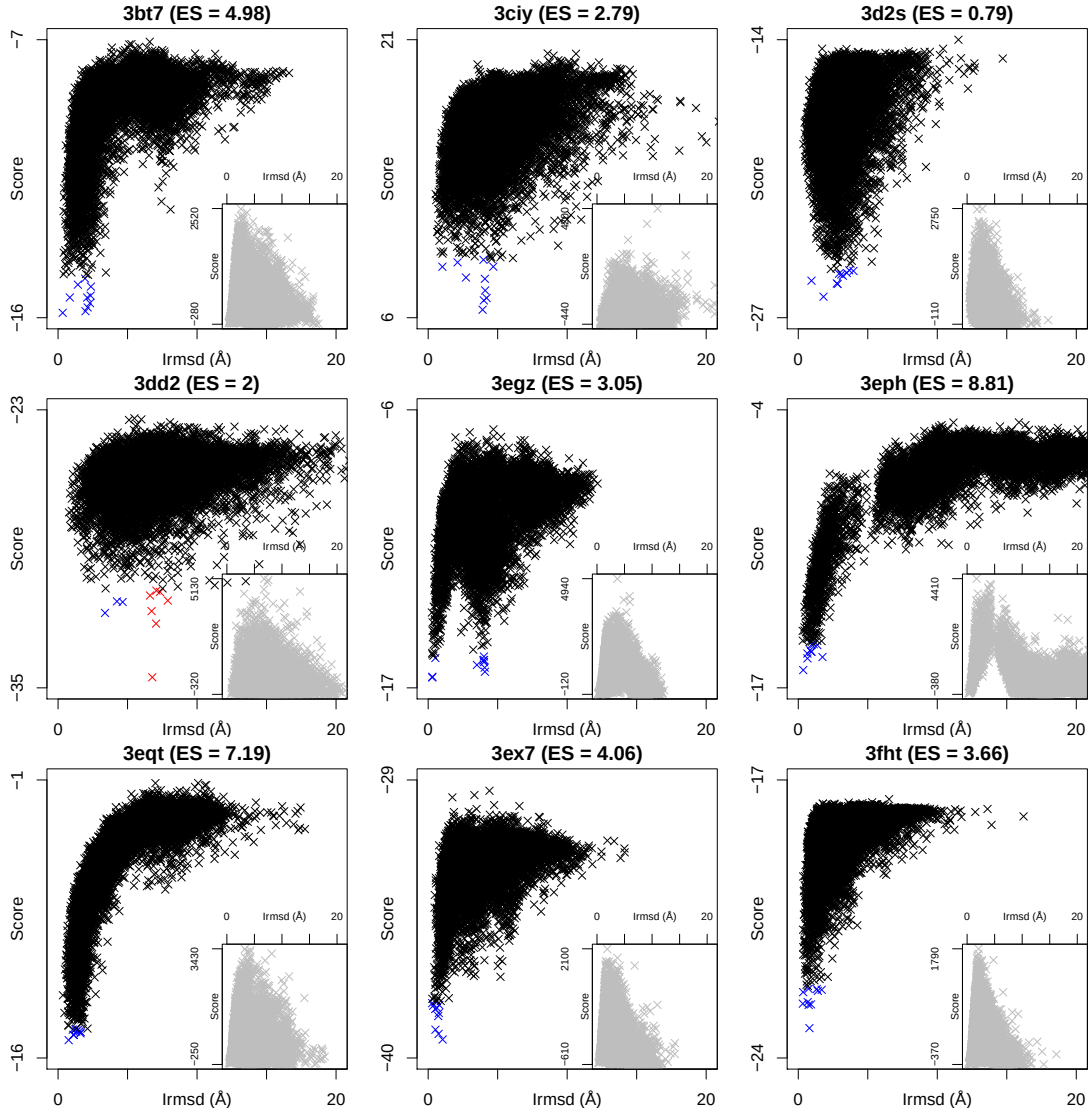


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

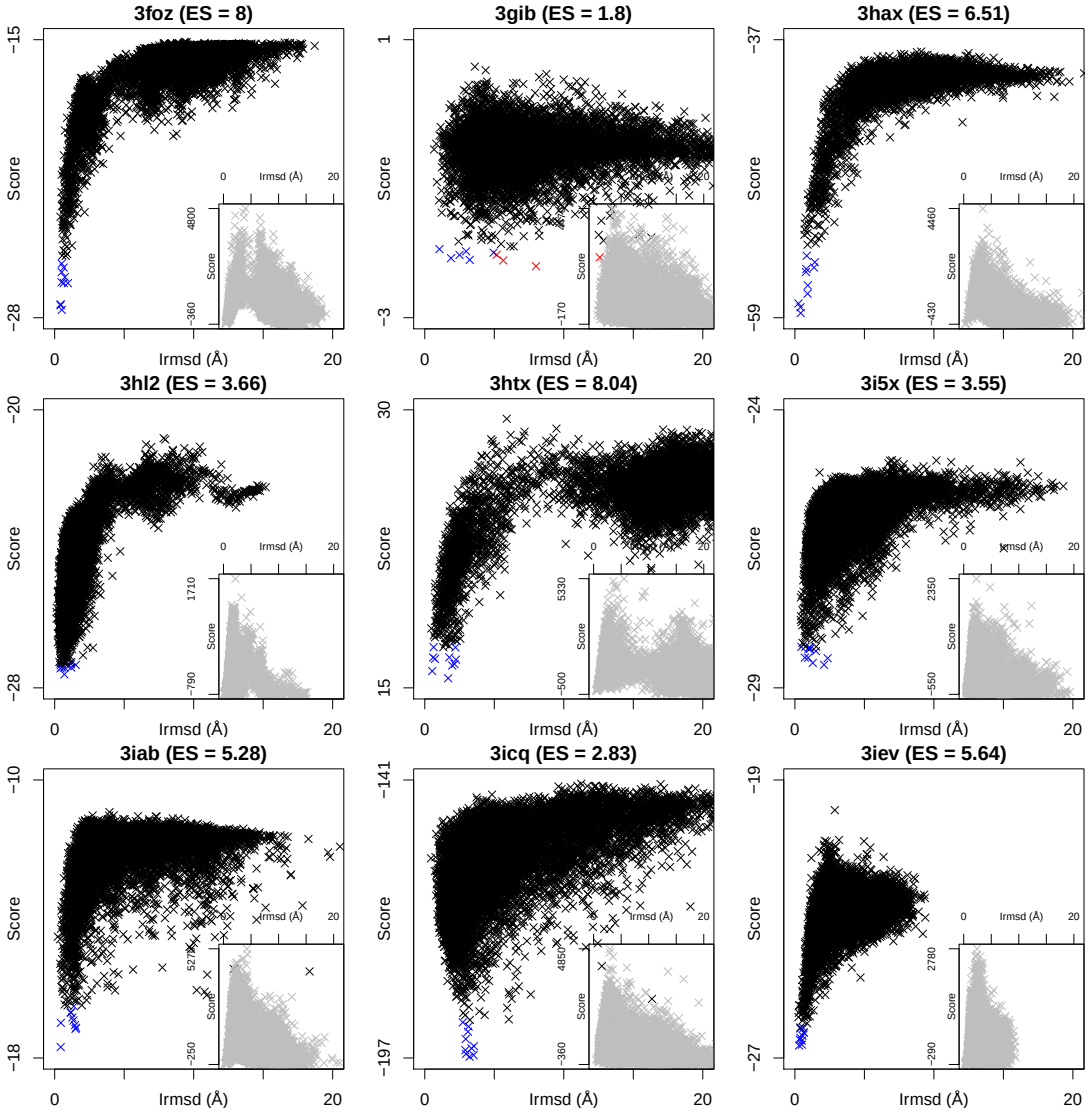


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

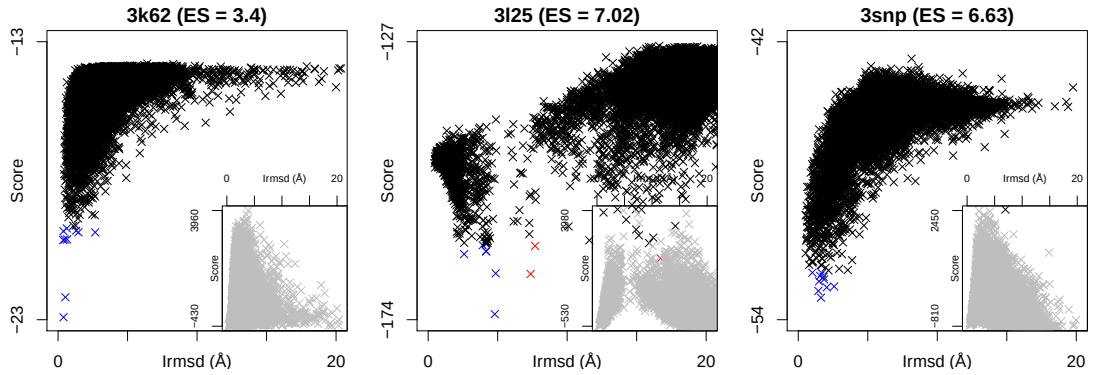


FIGURE S1 (cont.) – Diagramme par complexe de l'énergie en fonction du IRMSD (Evs-RMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

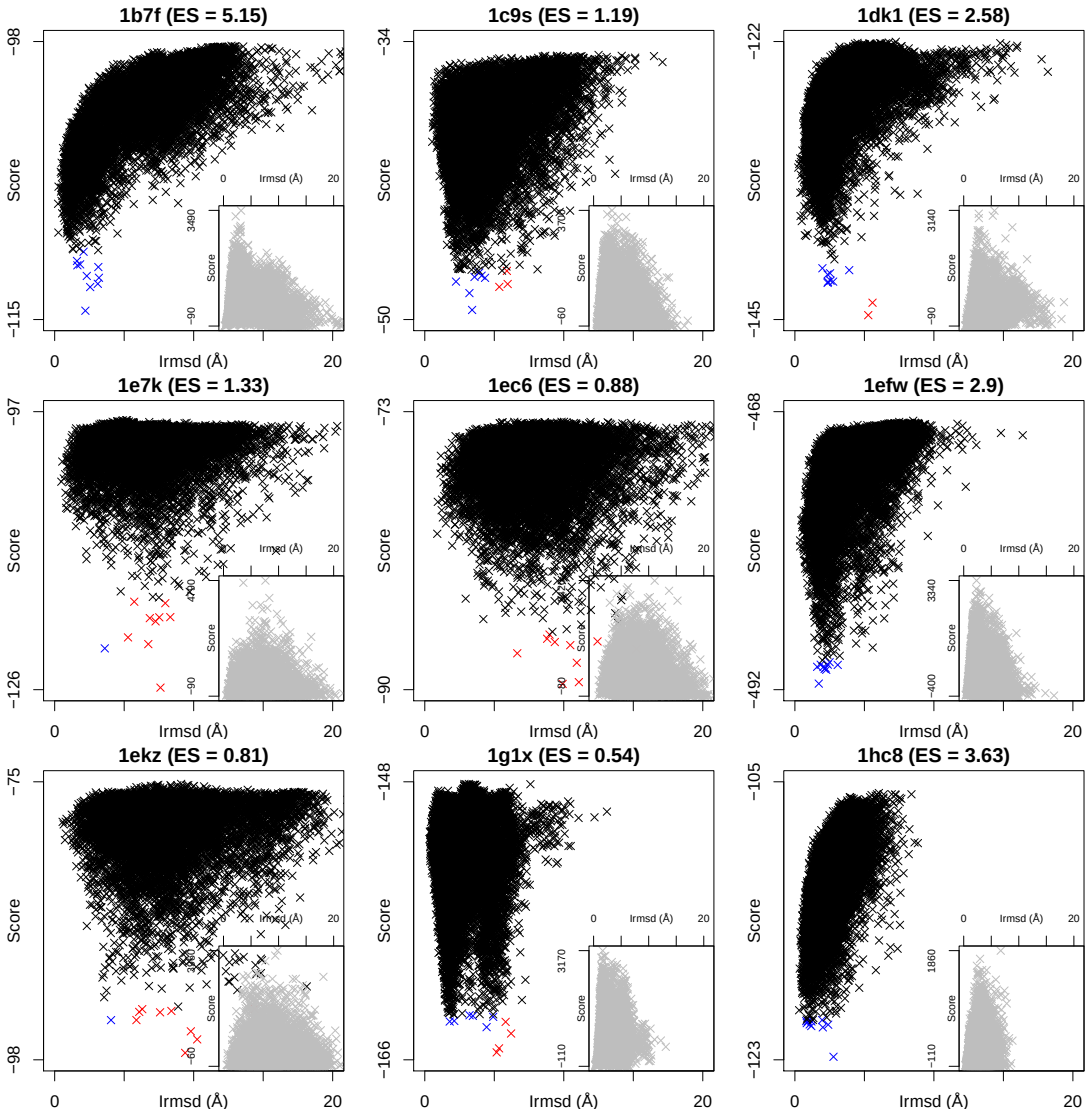


FIGURE S2 – Diagramme par complexe des *Benchmarks* I et II de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

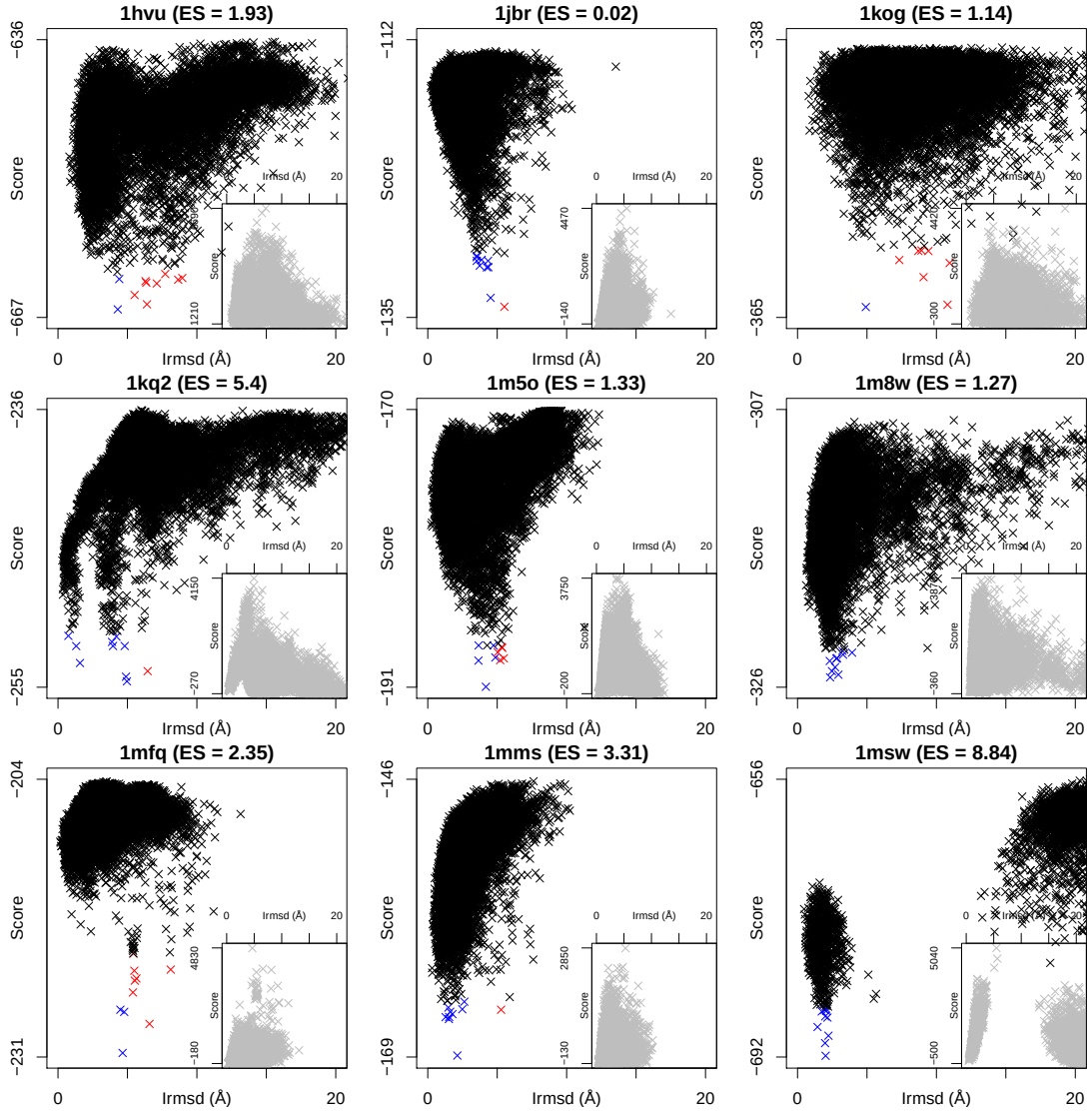


FIGURE S2 (cont.) – Diagramme par complexe des *Benchmarks* I et II de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

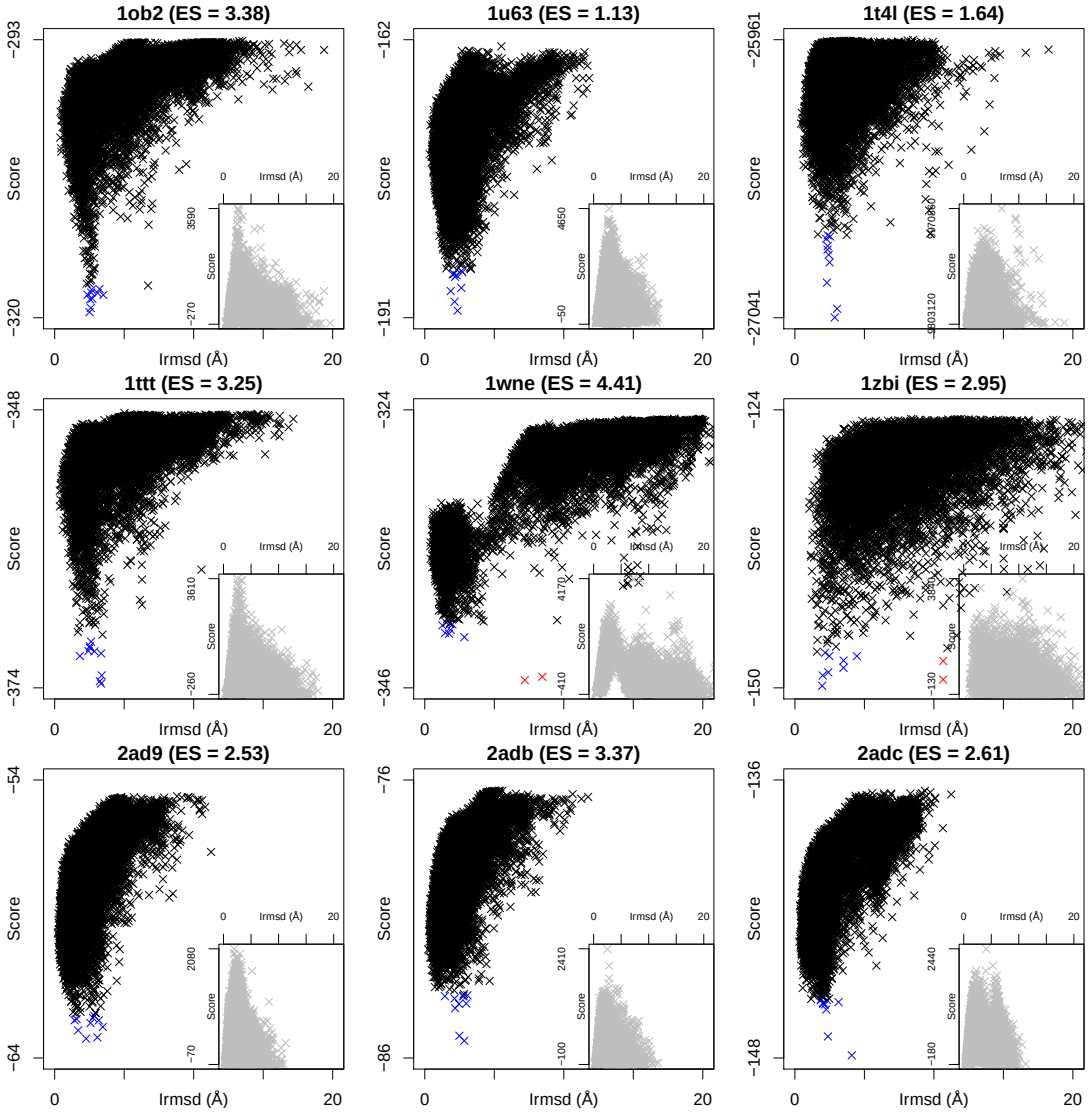


FIGURE S2 (cont.) – Diagramme par complexe des *Benchmarks* I et II de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

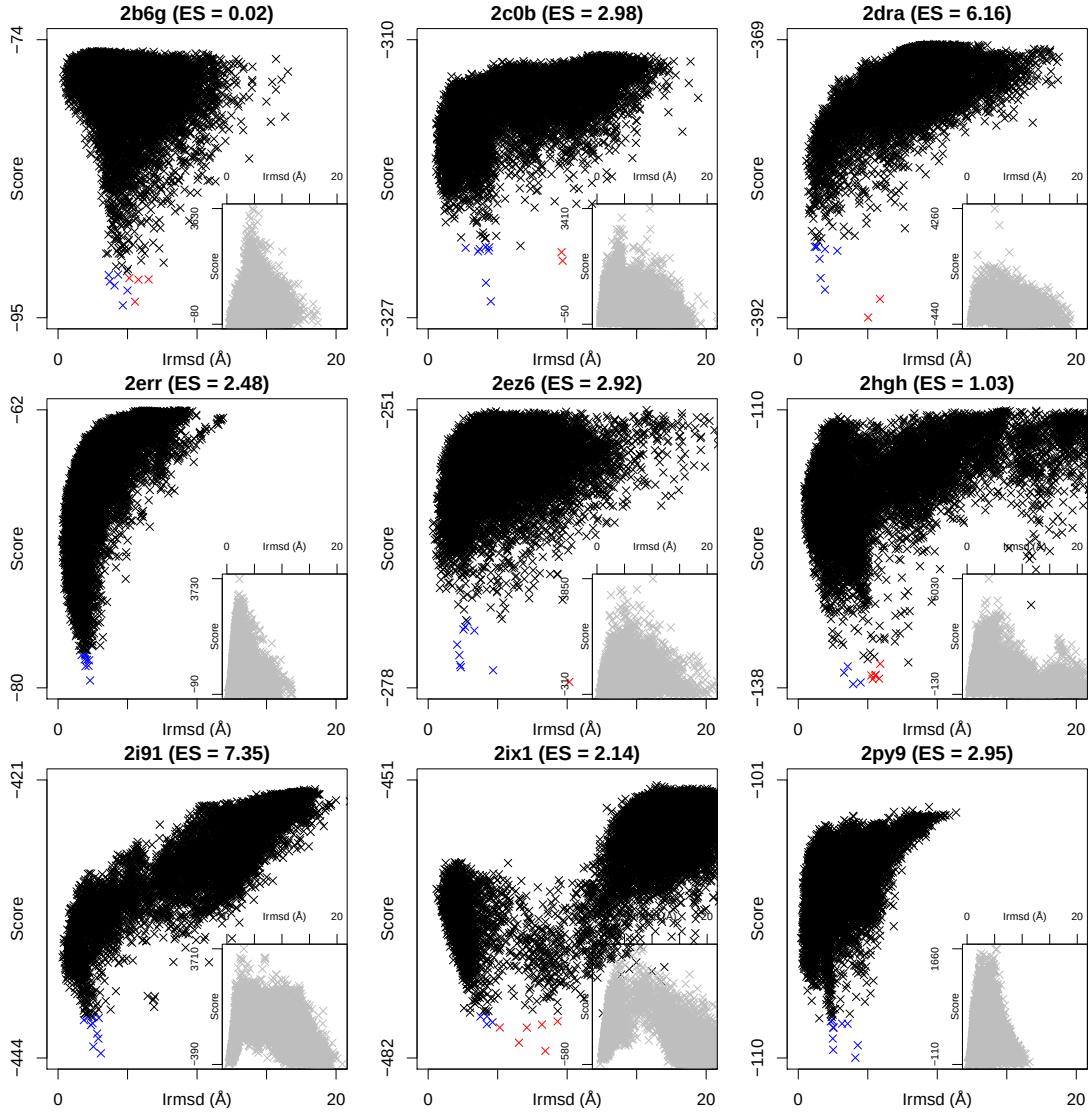


FIGURE S2 (cont.) – Diagramme par complexe des *Benchmarks* I et II de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

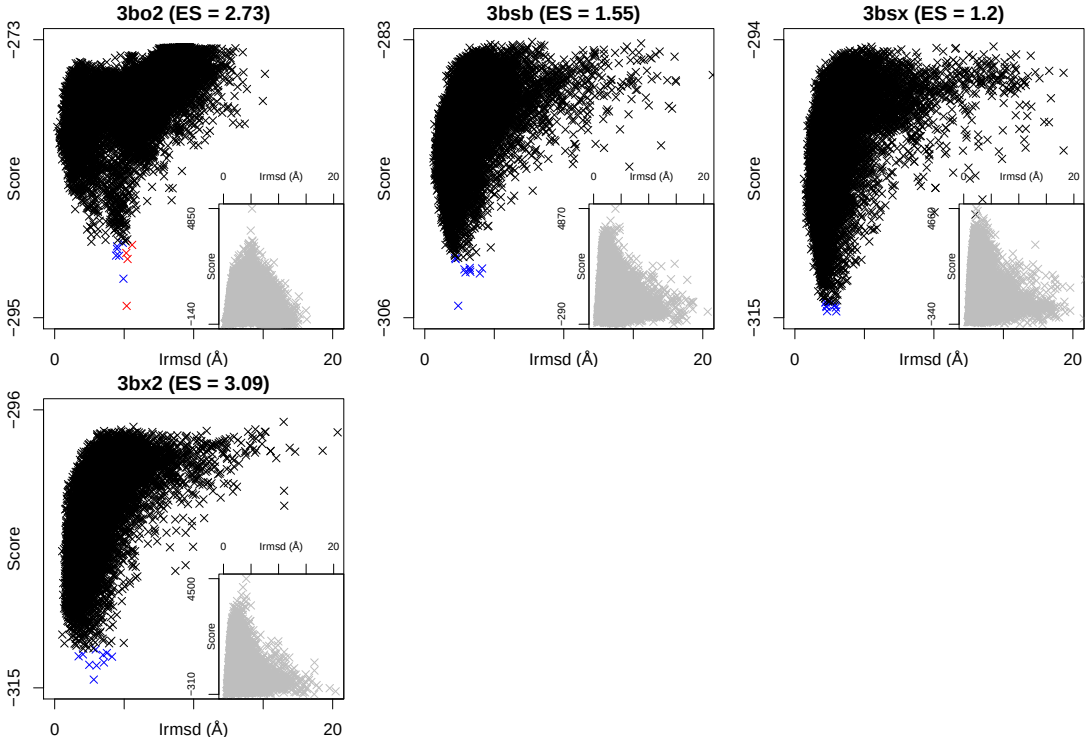


FIGURE S2 (cont.) – Diagramme par complexe des *Benchmarks* I et II de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

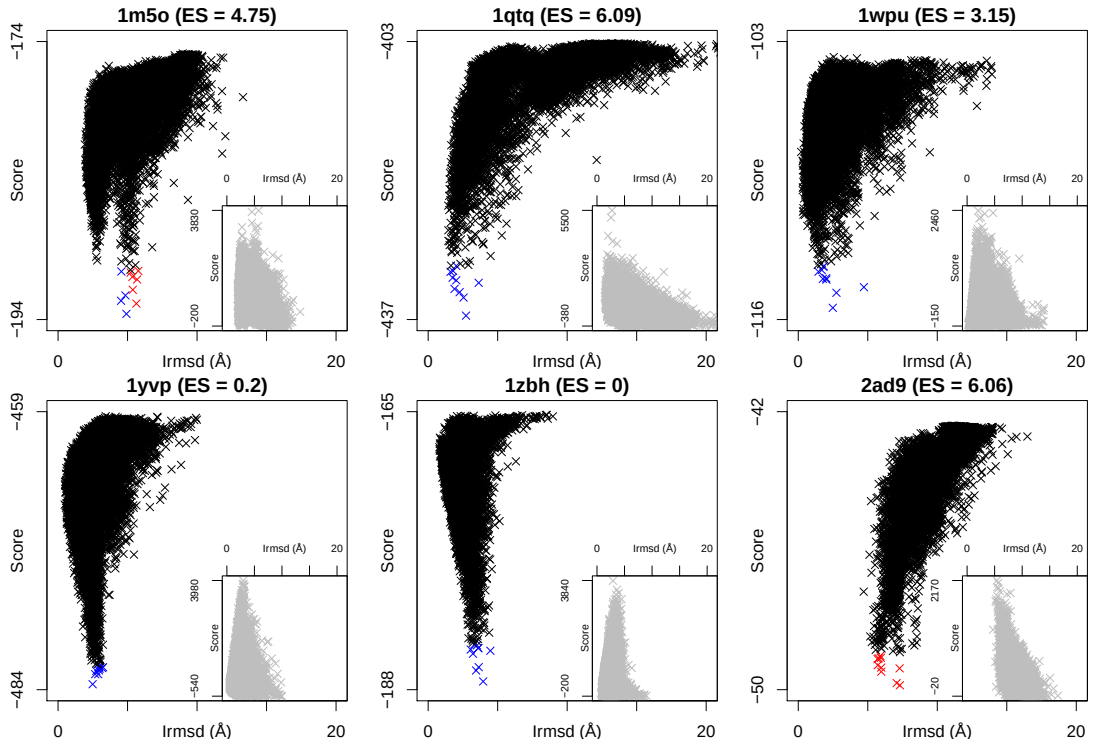


FIGURE S3 – Diagramme par complexe des 6 complexes avec protéine non liée de l'énergie en fonction du IRMSD (EvsRMS). Chaque candidat est positionné (par une croix) sur le diagramme selon son énergie et son IRMSD. Les 10 premiers candidats en énergie sont indiqués comme presque-natifs (en bleu) ou leurres (en rouge). La fonction de score utilisée est celle apprise par ROGER en *leave-"one-pdb"-out* pour les grands diagrammes et celle disponible par défaut dans RosettaDock dans les encarts en bas à droite de chaque diagramme (candidats indiqués en gris).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1asy	0.72	0.75	1.00	0.99	0.84	0.86	0.72	0.76	1.00	0.99	0.00	0.16
1av6	0.88	0.88	1.00	1.00	0.94	0.94	0.88	0.88	1.00	1.00	0.00	0.01
1b23	0.50	0.66	1.00	0.90	0.67	0.76	0.50	0.72	1.00	0.90	0.00	0.54
1c0a	0.26	0.71	1.00	0.78	0.41	0.74	0.26	0.86	1.00	0.78	0.00	0.89
1ddl	0.28	0.34	1.00	0.81	0.44	0.48	0.28	0.51	1.00	0.81	0.00	0.39
1dfu	0.92	0.96	1.00	0.99	0.96	0.97	0.92	0.95	1.00	0.99	0.00	0.44
1di2	0.76	0.84	1.00	0.94	0.86	0.88	0.76	0.82	1.00	0.94	0.00	0.45
1e8o	0.34	0.35	1.00	0.95	0.50	0.51	0.34	0.38	1.00	0.95	0.00	0.09
1f7u	0.41	0.76	1.00	0.85	0.58	0.80	0.41	0.83	1.00	0.85	0.00	0.81
1feu	0.91	0.92	1.00	0.99	0.95	0.95	0.91	0.91	1.00	0.99	0.00	0.10
1ffy	0.32	0.58	0.98	0.77	0.48	0.66	0.33	0.76	0.98	0.77	0.04	0.75
1fxl	0.51	0.86	1.00	0.87	0.68	0.87	0.51	0.86	1.00	0.87	0.00	0.86
1gtf	0.96	0.96	1.00	1.00	0.98	0.98	0.96	0.96	1.00	1.00	0.00	0.02
1h3e	0.59	0.69	1.00	0.94	0.74	0.79	0.59	0.71	1.00	0.94	0.00	0.38
1h4s	0.12	0.19	1.00	0.52	0.21	0.27	0.12	0.67	1.00	0.52	0.00	0.69
1hq1	0.28	0.28	0.89	0.97	0.42	0.43	0.36	0.33	0.89	0.97	0.17	0.10
1j1u	0.83	0.83	1.00	1.00	0.91	0.91	0.83	0.83	1.00	1.00	0.00	0.00
1j2b	0.13	0.64	1.00	0.81	0.23	0.71	0.13	0.91	1.00	0.81	0.00	0.93
1jbs	0.25	0.31	1.00	0.79	0.40	0.45	0.25	0.50	1.00	0.79	0.00	0.40
1jid	0.94	0.94	1.00	1.00	0.97	0.97	0.94	0.94	1.00	1.00	0.00	0.08
1k8w	0.39	0.72	1.00	0.86	0.56	0.78	0.39	0.81	1.00	0.86	0.00	0.79
1knz	0.31	0.84	1.00	0.89	0.47	0.86	0.31	0.91	1.00	0.89	0.00	0.92
1lng	0.73	0.76	1.00	0.98	0.84	0.86	0.73	0.76	1.00	0.98	0.00	0.16
1m8v	0.70	0.70	1.00	1.00	0.82	0.83	0.70	0.70	1.00	1.00	0.00	0.01
1m8x	0.84	0.84	1.00	1.00	0.91	0.91	0.84	0.84	1.00	1.00	0.00	0.01
1mzp	0.16	0.32	1.00	0.52	0.27	0.39	0.16	0.75	1.00	0.52	0.00	0.79
1n35	0.07	0.57	1.00	0.83	0.14	0.68	0.08	0.94	1.00	0.83	0.00	0.95
1n78	0.44	0.83	1.00	0.88	0.61	0.85	0.44	0.87	1.00	0.88	0.00	0.86
1ooa	0.87	0.88	1.00	1.00	0.93	0.94	0.87	0.88	1.00	1.00	0.00	0.14
1pgl	0.93	0.94	1.00	1.00	0.97	0.97	0.93	0.94	1.00	1.00	0.00	0.10
1q2r	0.46	0.73	1.00	0.92	0.63	0.82	0.46	0.81	1.00	0.92	0.00	0.71
1qf6	0.12	0.67	1.00	0.80	0.21	0.73	0.12	0.93	1.00	0.80	0.00	0.95
1qtq	0.30	0.72	1.00	0.77	0.46	0.74	0.30	0.84	1.00	0.77	0.00	0.87
1r3e	0.34	0.60	1.00	0.74	0.50	0.66	0.34	0.75	1.00	0.74	0.00	0.76
1r9f	0.94	0.94	1.00	1.00	0.97	0.97	0.94	0.94	1.00	1.00	0.00	0.04
1sds	0.43	0.42	0.96	1.00	0.59	0.59	0.44	0.43	0.96	1.00	0.07	0.02
1ser	0.30	0.66	1.00	0.87	0.46	0.75	0.30	0.83	1.00	0.87	0.00	0.81
1si3	0.77	0.95	1.00	0.95	0.87	0.95	0.77	0.92	1.00	0.95	0.00	0.84
1t0k	0.38	0.37	0.98	1.00	0.54	0.54	0.39	0.37	0.98	1.00	0.03	0.00
1tfw	0.02	0.71	1.00	0.96	0.04	0.82	0.02	0.99	1.00	0.96	0.00	0.99
1u0b	0.26	0.58	1.00	0.77	0.41	0.66	0.26	0.79	1.00	0.77	0.00	0.80
1un6	0.40	0.67	1.00	0.73	0.57	0.70	0.40	0.75	1.00	0.73	0.00	0.76
1uvj	0.21	0.72	1.00	0.92	0.34	0.81	0.21	0.91	1.00	0.92	0.00	0.91
1vfg	0.93	0.93	1.00	1.00	0.96	0.96	0.93	0.93	1.00	1.00	0.00	0.00
1wpu	0.87	0.88	1.00	1.00	0.93	0.93	0.87	0.88	1.00	1.00	0.00	0.05
1wsu	0.48	0.51	1.00	0.95	0.65	0.67	0.48	0.54	1.00	0.95	0.00	0.15
1wz2	0.25	0.51	1.00	0.78	0.39	0.62	0.25	0.76	1.00	0.78	0.00	0.76

Résultats globaux des 120 complexes de la PRIDB sous le seuil du meilleur Fscore : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1yvp	0.94	0.94	1.00	1.00	0.97	0.97	0.94	0.94	1.00	1.00	0.00	0.01
1zbh	0.38	0.47	1.00	0.90	0.55	0.61	0.38	0.57	1.00	0.90	0.00	0.37
2a8v	0.48	0.54	1.00	0.87	0.65	0.67	0.48	0.59	1.00	0.87	0.00	0.33
2anr	0.29	0.36	1.00	0.76	0.45	0.49	0.29	0.54	1.00	0.76	0.00	0.44
2asb	0.55	0.68	1.00	0.97	0.71	0.80	0.55	0.73	1.00	0.97	0.00	0.45
2az0	0.14	0.32	0.99	0.58	0.24	0.41	0.14	0.77	0.99	0.58	0.01	0.81
2azx	0.72	0.77	1.00	0.97	0.84	0.86	0.72	0.77	1.00	0.97	0.00	0.27
2b3j	0.72	0.88	1.00	0.94	0.84	0.91	0.72	0.87	1.00	0.94	0.00	0.68
2bgg	0.54	0.81	1.00	0.87	0.70	0.84	0.54	0.82	1.00	0.87	0.00	0.76
2bh2	0.23	0.76	1.00	0.70	0.38	0.73	0.23	0.88	1.00	0.70	0.00	0.93
2bte	0.73	0.80	1.00	0.93	0.84	0.86	0.73	0.78	1.00	0.93	0.00	0.39
2bu1	0.39	0.43	1.00	0.91	0.56	0.59	0.39	0.50	1.00	0.91	0.00	0.24
2bx2	0.35	0.51	0.99	0.89	0.51	0.65	0.35	0.67	0.99	0.89	0.01	0.56
2ct8	0.58	0.72	1.00	0.90	0.74	0.80	0.58	0.74	1.00	0.90	0.00	0.53
2czj	0.91	0.91	1.00	1.00	0.95	0.96	0.91	0.91	1.00	1.00	0.00	0.01
2d6f	0.85	0.85	1.00	1.00	0.92	0.92	0.85	0.85	1.00	1.00	0.01	0.00
2der	0.35	0.83	1.00	0.79	0.52	0.81	0.35	0.87	1.00	0.79	0.00	0.91
2du3	0.92	0.92	1.00	1.00	0.96	0.96	0.92	0.92	1.00	1.00	0.00	0.01
2e9t	0.17	0.93	1.00	0.98	0.29	0.95	0.17	0.98	1.00	0.98	0.00	0.99
2f8k	0.50	0.50	1.00	0.99	0.67	0.67	0.50	0.51	1.00	0.99	0.00	0.02
2f8s	0.68	0.69	1.00	0.99	0.81	0.82	0.68	0.69	1.00	0.99	0.00	0.05
2fk6	0.82	0.83	1.00	0.99	0.90	0.90	0.82	0.83	1.00	0.99	0.00	0.09
2fmt	0.27	0.53	1.00	0.88	0.42	0.66	0.27	0.76	1.00	0.88	0.00	0.71
2gic	0.40	0.69	1.00	0.90	0.57	0.78	0.40	0.80	1.00	0.90	0.00	0.74
2gje	0.72	0.78	1.00	0.95	0.84	0.85	0.72	0.76	1.00	0.95	0.00	0.28
2gjw	0.78	0.85	1.00	0.97	0.88	0.91	0.78	0.84	1.00	0.97	0.00	0.41
2gtt	0.24	0.51	0.98	0.87	0.39	0.64	0.26	0.76	0.98	0.87	0.03	0.73
2gxb	0.47	0.49	1.00	0.93	0.64	0.64	0.47	0.52	1.00	0.93	0.00	0.15
2hw8	0.73	0.81	1.00	0.97	0.84	0.88	0.73	0.81	1.00	0.97	0.00	0.38
2i82	0.69	0.78	1.00	0.95	0.82	0.85	0.69	0.78	1.00	0.95	0.00	0.39
2iy5	0.09	0.47	1.00	0.83	0.17	0.60	0.10	0.90	1.00	0.83	0.01	0.91
2jlv	0.46	0.71	0.99	0.94	0.63	0.81	0.47	0.80	0.99	0.94	0.04	0.68
2nqp	0.27	0.35	1.00	0.82	0.42	0.49	0.27	0.54	1.00	0.82	0.01	0.44
2nug	0.06	1.00	0.58	1.00	0.11	1.00	0.66	1.00	0.58	1.00	0.66	1.00
2ozb	0.37	0.66	1.00	0.76	0.54	0.71	0.37	0.77	1.00	0.76	0.00	0.77
2pjp	0.30	0.39	1.00	0.83	0.46	0.53	0.30	0.56	1.00	0.83	0.00	0.45
2po1	0.92	0.91	1.00	1.00	0.96	0.96	0.92	0.91	1.00	1.00	0.03	0.00
2qux	0.19	0.28	1.00	0.77	0.33	0.41	0.20	0.57	1.00	0.77	0.01	0.52
2r7r	0.42	0.83	1.00	0.94	0.59	0.88	0.42	0.89	1.00	0.94	0.00	0.86
2r8s	0.73	0.73	1.00	1.00	0.84	0.84	0.73	0.73	1.00	1.00	0.00	0.04
2vnu	0.25	0.69	1.00	0.96	0.40	0.80	0.25	0.88	1.00	0.96	0.00	0.86
2voo	0.76	0.76	1.00	1.00	0.86	0.86	0.76	0.76	1.00	1.00	0.00	0.00
2w2h	0.55	0.63	1.00	0.91	0.71	0.75	0.55	0.66	1.00	0.91	0.00	0.35
2wj8	0.24	0.82	1.00	0.93	0.39	0.88	0.24	0.94	1.00	0.93	0.00	0.94
2z2q	0.59	0.60	1.00	0.99	0.74	0.75	0.59	0.60	1.00	0.99	0.00	0.05
2zi0	0.79	0.89	1.00	0.98	0.88	0.93	0.79	0.89	1.00	0.98	0.00	0.55
2zko	0.17	0.24	0.52	0.62	0.26	0.34	0.61	0.68	0.52	0.62	0.62	0.69

Résultats globaux des 120 complexes de la PRIDB sous le seuil du meilleur Fscore : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
2zni	0.27	0.52	1.00	0.71	0.42	0.60	0.27	0.75	1.00	0.71	0.00	0.76
2zue	0.49	0.77	1.00	0.88	0.66	0.82	0.49	0.81	1.00	0.88	0.00	0.74
2zzm	0.13	0.76	1.00	0.79	0.22	0.78	0.13	0.94	1.00	0.79	0.00	0.96
3a6p	0.61	0.88	0.56	0.90	0.58	0.89	0.84	0.96	0.56	0.90	0.91	0.97
3bso	0.25	0.77	0.99	0.94	0.40	0.85	0.27	0.92	0.99	0.94	0.04	0.91
3bt7	0.54	0.67	1.00	0.87	0.70	0.76	0.54	0.70	1.00	0.87	0.00	0.50
3ciy	0.56	0.58	1.00	0.98	0.72	0.73	0.56	0.58	1.00	0.98	0.00	0.07
3d2s	0.89	0.90	1.00	1.00	0.94	0.94	0.89	0.89	1.00	1.00	0.00	0.02
3dd2	0.25	0.30	1.00	0.79	0.40	0.43	0.25	0.48	1.00	0.79	0.00	0.38
3egz	0.52	0.82	1.00	0.91	0.69	0.87	0.52	0.85	1.00	0.91	0.00	0.79
3eph	0.10	0.93	1.00	0.95	0.18	0.94	0.10	0.99	1.00	0.95	0.00	0.99
3eqt	0.52	0.85	1.00	0.76	0.68	0.80	0.52	0.80	1.00	0.76	0.00	0.85
3ex7	0.75	0.87	1.00	0.96	0.86	0.91	0.75	0.86	1.00	0.96	0.00	0.59
3fht	0.83	0.92	1.00	0.95	0.90	0.93	0.83	0.89	1.00	0.95	0.00	0.59
3foz	0.13	0.68	1.00	0.92	0.24	0.78	0.13	0.93	1.00	0.92	0.00	0.93
3gib	0.23	0.40	1.00	0.64	0.38	0.49	0.23	0.69	1.00	0.64	0.00	0.71
3hax	0.10	0.76	1.00	0.60	0.19	0.67	0.10	0.94	1.00	0.60	0.00	0.98
3hl2	0.91	0.95	1.00	0.99	0.95	0.97	0.91	0.94	1.00	0.99	0.00	0.48
3htx	0.10	0.63	1.00	0.76	0.19	0.69	0.10	0.93	1.00	0.76	0.00	0.95
3i5x	0.68	0.79	1.00	0.96	0.81	0.87	0.68	0.80	1.00	0.96	0.00	0.47
3iab	0.48	0.74	1.00	0.85	0.65	0.79	0.48	0.78	1.00	0.85	0.00	0.73
3icq	0.51	0.67	1.00	0.89	0.68	0.76	0.51	0.72	1.00	0.89	0.00	0.53
3iev	0.79	0.79	1.00	1.00	0.88	0.88	0.79	0.79	1.00	1.00	0.00	0.01
3k62	0.85	0.85	1.00	1.00	0.92	0.92	0.85	0.85	1.00	1.00	0.00	0.03
3l25	0.12	0.92	1.00	0.96	0.22	0.94	0.12	0.98	1.00	0.96	0.00	0.99
3snp	0.32	0.63	1.00	0.75	0.48	0.68	0.32	0.78	1.00	0.75	0.00	0.79

TABLE S1 – Résultats globaux des 120 complexes de la PRIDB sous le seuil du meilleur Fscore : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1asy	0.70	0.90	0.00	0.00	0.00	0.00	0.28	0.28	0.00	0.00	1.00	1.00
1av6	0.50	1.00	0.00	0.00	0.00	0.00	0.12	0.12	0.00	0.00	1.00	1.00
1b23	0.40	1.00	0.00	0.00	0.00	0.00	0.50	0.50	0.00	0.00	1.00	1.00
1c0a	0.10	1.00	0.00	0.00	0.00	0.01	0.74	0.74	0.00	0.00	1.00	1.00
1ddl	0.20	0.40	0.00	0.00	0.00	0.00	0.72	0.72	0.00	0.00	1.00	1.00
1dfu	0.40	1.00	0.00	0.00	0.00	0.00	0.08	0.08	0.00	0.00	0.99	1.00
1di2	1.00	0.90	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00	1.00	1.00
1e8o	0.60	0.40	0.00	0.00	0.00	0.00	0.66	0.66	0.00	0.00	1.00	1.00
1f7u	0.40	1.00	0.00	0.00	0.00	0.00	0.59	0.59	0.00	0.00	1.00	1.00
1feu	0.40	1.00	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.00	0.99	1.00
1ffy	0.00	0.80	0.00	0.00	0.00	0.01	0.69	0.69	0.00	0.00	1.00	1.00
1fxl	0.00	1.00	0.00	0.00	0.00	0.00	0.49	0.49	0.00	0.00	1.00	1.00
1gtf	1.00	1.00	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.00	1.00	1.00
1h3e	0.00	0.80	0.00	0.00	0.00	0.00	0.41	0.41	0.00	0.00	1.00	1.00
1h4s	0.10	0.50	0.00	0.00	0.00	0.01	0.88	0.88	0.00	0.00	1.00	1.00
1hq1	0.60	0.00	0.00	0.00	0.00	0.00	0.74	0.74	0.00	0.00	1.00	1.00
1j1u	1.00	1.00	0.00	0.00	0.00	0.00	0.17	0.17	0.00	0.00	1.00	1.00
1j2b	0.00	0.90	0.00	0.01	0.00	0.01	0.87	0.87	0.00	0.01	1.00	1.00
1jbs	0.00	0.20	0.00	0.00	0.00	0.00	0.75	0.75	0.00	0.00	1.00	1.00
1jid	0.20	1.00	0.00	0.00	0.00	0.00	0.06	0.07	0.00	0.00	0.99	1.00
1k8w	0.10	1.00	0.00	0.00	0.00	0.01	0.61	0.61	0.00	0.00	1.00	1.00
1knz	0.00	0.50	0.00	0.00	0.00	0.00	0.69	0.69	0.00	0.00	1.00	1.00
1lng	0.60	0.90	0.00	0.00	0.00	0.00	0.27	0.27	0.00	0.00	1.00	1.00
1m8v	1.00	0.90	0.00	0.00	0.00	0.00	0.30	0.30	0.00	0.00	1.00	1.00
1m8x	0.70	1.00	0.00	0.00	0.00	0.00	0.16	0.16	0.00	0.00	1.00	1.00
1mzp	0.00	0.60	0.00	0.00	0.00	0.01	0.84	0.84	0.00	0.00	1.00	1.00
1n35	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.93	0.00	0.00	1.00	1.00
1n78	0.00	1.00	0.00	0.00	0.00	0.00	0.56	0.56	0.00	0.00	1.00	1.00
1ooa	0.60	0.90	0.00	0.00	0.00	0.00	0.13	0.13	0.00	0.00	1.00	1.00
1pgl	1.00	1.00	0.00	0.00	0.00	0.00	0.07	0.07	0.00	0.00	1.00	1.00
1q2r	0.00	1.00	0.00	0.00	0.00	0.00	0.54	0.54	0.00	0.00	1.00	1.00
1qf6	0.00	1.00	0.00	0.01	0.00	0.02	0.88	0.89	0.00	0.01	1.00	1.00
1qtq	0.00	1.00	0.00	0.00	0.00	0.01	0.70	0.70	0.00	0.00	1.00	1.00
1r3e	0.00	0.90	0.00	0.00	0.00	0.01	0.66	0.67	0.00	0.00	1.00	1.00
1r9f	1.00	1.00	0.00	0.00	0.00	0.00	0.07	0.07	0.00	0.00	1.00	1.00
1sds	0.50	0.40	0.00	0.00	0.00	0.00	0.58	0.58	0.00	0.00	1.00	1.00
1ser	0.00	0.60	0.00	0.00	0.00	0.00	0.70	0.70	0.00	0.00	1.00	1.00
1si3	0.00	1.00	0.00	0.00	0.00	0.00	0.23	0.23	0.00	0.00	1.00	1.00
1t0k	0.80	0.00	0.00	0.00	0.00	0.00	0.63	0.63	0.00	0.00	1.00	1.00
1tfw	0.00	0.40	0.00	0.02	0.00	0.03	0.98	0.98	0.00	0.02	1.00	1.00
1u0b	0.00	1.00	0.00	0.00	0.00	0.01	0.74	0.74	0.00	0.00	1.00	1.00
1un6	0.00	0.90	0.00	0.00	0.00	0.00	0.60	0.61	0.00	0.00	1.00	1.00
1uvj	0.00	0.40	0.00	0.00	0.00	0.00	0.79	0.79	0.00	0.00	1.00	1.00
1vfg	0.90	0.90	0.00	0.00	0.00	0.00	0.07	0.07	0.00	0.00	1.00	1.00
1wpu	1.00	1.00	0.00	0.00	0.00	0.00	0.13	0.13	0.00	0.00	1.00	1.00
1wsu	0.00	0.60	0.00	0.00	0.00	0.00	0.52	0.52	0.00	0.00	1.00	1.00
1wz2	0.00	0.90	0.00	0.00	0.00	0.01	0.75	0.76	0.00	0.00	1.00	1.00

Résultats globaux des 120 complexes de la PRIDB sous le seuil du top10 des candidats : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1yvp	0.80	1.00	0.00	0.00	0.00	0.00	0.06	0.06	0.00	0.00	1.00	1.00
1zbh	0.20	0.80	0.00	0.00	0.00	0.00	0.62	0.62	0.00	0.00	1.00	1.00
2a8v	0.20	0.50	0.00	0.00	0.00	0.00	0.52	0.52	0.00	0.00	1.00	1.00
2anr	0.70	0.10	0.00	0.00	0.00	0.00	0.71	0.71	0.00	0.00	1.00	1.00
2asb	0.20	1.00	0.00	0.00	0.00	0.00	0.45	0.45	0.00	0.00	1.00	1.00
2az0	0.00	0.40	0.00	0.00	0.00	0.01	0.86	0.86	0.00	0.00	1.00	1.00
2azx	0.00	0.90	0.00	0.00	0.00	0.00	0.28	0.28	0.00	0.00	1.00	1.00
2b3j	0.00	1.00	0.00	0.00	0.00	0.00	0.28	0.28	0.00	0.00	1.00	1.00
2bgg	0.00	1.00	0.00	0.00	0.00	0.00	0.46	0.46	0.00	0.00	1.00	1.00
2bh2	0.00	1.00	0.00	0.00	0.00	0.01	0.77	0.77	0.00	0.00	1.00	1.00
2bte	0.70	1.00	0.00	0.00	0.00	0.00	0.28	0.28	0.00	0.00	1.00	1.00
2bu1	0.50	0.30	0.00	0.00	0.00	0.00	0.61	0.61	0.00	0.00	1.00	1.00
2bx2	0.00	0.80	0.00	0.00	0.00	0.00	0.65	0.66	0.00	0.00	1.00	1.00
2ct8	0.30	1.00	0.00	0.00	0.00	0.00	0.42	0.42	0.00	0.00	1.00	1.00
2czj	1.00	1.00	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.00	1.00	1.00
2d6f	1.00	0.50	0.00	0.00	0.00	0.00	0.16	0.15	0.00	0.00	1.00	1.00
2der	0.00	0.90	0.00	0.00	0.00	0.01	0.65	0.65	0.00	0.00	1.00	1.00
2du3	0.90	1.00	0.00	0.00	0.00	0.00	0.08	0.09	0.00	0.00	1.00	1.00
2e9t	0.00	0.90	0.00	0.01	0.00	0.01	0.83	0.83	0.00	0.01	1.00	1.00
2f8k	0.90	0.50	0.00	0.00	0.00	0.00	0.50	0.50	0.00	0.00	1.00	1.00
2f8s	0.60	1.00	0.00	0.00	0.00	0.00	0.32	0.32	0.00	0.00	1.00	1.00
2fk6	1.00	0.90	0.00	0.00	0.00	0.00	0.18	0.18	0.00	0.00	1.00	1.00
2fmt	0.00	1.00	0.00	0.00	0.00	0.01	0.73	0.73	0.00	0.00	1.00	1.00
2gic	0.00	0.60	0.00	0.00	0.00	0.00	0.60	0.60	0.00	0.00	1.00	1.00
2gje	0.60	1.00	0.00	0.00	0.00	0.00	0.28	0.28	0.00	0.00	1.00	1.00
2gjw	0.80	1.00	0.00	0.00	0.00	0.00	0.22	0.22	0.00	0.00	1.00	1.00
2gtt	0.00	0.40	0.00	0.00	0.00	0.00	0.76	0.76	0.00	0.00	1.00	1.00
2gxb	0.00	0.40	0.00	0.00	0.00	0.00	0.53	0.53	0.00	0.00	1.00	1.00
2hw8	0.90	0.90	0.00	0.00	0.00	0.00	0.27	0.27	0.00	0.00	1.00	1.00
2i82	0.60	1.00	0.00	0.00	0.00	0.00	0.31	0.31	0.00	0.00	1.00	1.00
2iy5	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.91	0.00	0.00	1.00	1.00
2jlv	0.00	0.10	0.00	0.00	0.00	0.00	0.55	0.55	0.00	0.00	1.00	1.00
2nqp	0.00	0.20	0.00	0.00	0.00	0.00	0.73	0.74	0.00	0.00	1.00	1.00
2nug	0.00	1.00	0.00	0.03	0.00	0.05	0.96	0.96	0.00	0.03	1.00	1.00
2ozb	0.10	1.00	0.00	0.00	0.00	0.01	0.63	0.63	0.00	0.00	1.00	1.00
2pjp	0.10	0.60	0.00	0.00	0.00	0.00	0.70	0.70	0.00	0.00	1.00	1.00
2po1	1.00	0.30	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.00	1.00	0.99
2qux	0.10	0.00	0.00	0.00	0.00	0.00	0.81	0.81	0.00	0.00	1.00	1.00
2r7r	0.00	0.70	0.00	0.00	0.00	0.00	0.58	0.58	0.00	0.00	1.00	1.00
2r8s	0.80	0.90	0.00	0.00	0.00	0.00	0.27	0.27	0.00	0.00	1.00	1.00
2vnu	0.00	0.20	0.00	0.00	0.00	0.00	0.75	0.75	0.00	0.00	1.00	1.00
2voo	1.00	1.00	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00	1.00	1.00
2w2h	0.20	1.00	0.00	0.00	0.00	0.00	0.45	0.45	0.00	0.00	1.00	1.00
2wj8	0.00	0.80	0.00	0.00	0.00	0.01	0.76	0.76	0.00	0.00	1.00	1.00
2z2q	0.30	0.60	0.00	0.00	0.00	0.00	0.41	0.41	0.00	0.00	1.00	1.00
2zi0	0.00	0.60	0.00	0.00	0.00	0.00	0.21	0.21	0.00	0.00	1.00	1.00
2zko	0.00	0.10	0.00	0.00	0.00	0.00	0.87	0.87	0.00	0.00	1.00	1.00

Résultats globaux des 120 complexes de la PRIDB sous le seuil du top10 des candidats : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
2zni	0.00	1.00	0.00	0.00	0.00	0.01	0.73	0.73	0.00	0.00	1.00	1.00
2zue	0.00	1.00	0.00	0.00	0.00	0.00	0.51	0.51	0.00	0.00	1.00	1.00
2zzm	0.00	1.00	0.00	0.01	0.00	0.02	0.87	0.88	0.00	0.01	1.00	1.00
3a6p	0.00	0.70	0.00	0.00	0.00	0.01	0.80	0.80	0.00	0.00	1.00	1.00
3bso	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.75	0.00	0.00	1.00	1.00
3bt7	0.10	1.00	0.00	0.00	0.00	0.00	0.46	0.46	0.00	0.00	1.00	1.00
3ciy	0.40	0.30	0.00	0.00	0.00	0.00	0.44	0.44	0.00	0.00	1.00	1.00
3d2s	0.60	1.00	0.00	0.00	0.00	0.00	0.11	0.11	0.00	0.00	1.00	1.00
3dd2	0.00	0.30	0.00	0.00	0.00	0.00	0.75	0.75	0.00	0.00	1.00	1.00
3egz	0.60	0.90	0.00	0.00	0.00	0.00	0.48	0.48	0.00	0.00	1.00	1.00
3eph	0.00	1.00	0.00	0.01	0.00	0.02	0.90	0.90	0.00	0.01	1.00	1.00
3eqt	0.20	1.00	0.00	0.00	0.00	0.00	0.48	0.48	0.00	0.00	1.00	1.00
3ex7	0.00	1.00	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00	1.00	1.00
3fht	0.00	1.00	0.00	0.00	0.00	0.00	0.17	0.18	0.00	0.00	0.99	1.00
3foz	0.00	1.00	0.00	0.01	0.00	0.01	0.86	0.87	0.00	0.01	1.00	1.00
3gib	0.00	0.50	0.00	0.00	0.00	0.00	0.77	0.77	0.00	0.00	1.00	1.00
3hax	0.10	1.00	0.00	0.01	0.00	0.02	0.90	0.90	0.00	0.01	1.00	1.00
3hl2	0.00	1.00	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.00	0.99	1.00
3htx	0.00	0.90	0.00	0.01	0.00	0.02	0.89	0.90	0.00	0.01	1.00	1.00
3i5x	0.00	1.00	0.00	0.00	0.00	0.00	0.32	0.32	0.00	0.00	1.00	1.00
3iab	0.00	1.00	0.00	0.00	0.00	0.00	0.52	0.52	0.00	0.00	1.00	1.00
3icq	0.00	0.90	0.00	0.00	0.00	0.00	0.49	0.49	0.00	0.00	1.00	1.00
3iev	0.80	1.00	0.00	0.00	0.00	0.00	0.21	0.21	0.00	0.00	1.00	1.00
3k62	0.90	1.00	0.00	0.00	0.00	0.00	0.15	0.15	0.00	0.00	1.00	1.00
3l25	0.00	0.90	0.00	0.01	0.00	0.01	0.88	0.88	0.00	0.01	1.00	1.00
3snp	0.10	1.00	0.00	0.00	0.00	0.01	0.68	0.68	0.00	0.00	1.00	1.00

TABLE S2 – Résultats globaux des 120 complexes de la PRIDB sous le seuil du top10 des candidats : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1b7f	0.36	0.75	1.00	0.73	0.53	0.74	0.36	0.82	1.00	0.73	0.00	0.87
1c9s	0.59	0.64	1.00	0.94	0.74	0.77	0.59	0.66	1.00	0.94	0.00	0.25
1dk1	0.77	0.81	1.00	0.97	0.87	0.88	0.77	0.80	1.00	0.97	0.00	0.25
1e7k	0.34	0.34	0.98	1.00	0.50	0.50	0.35	0.34	0.98	1.00	0.04	0.00
1ec6	0.18	0.22	0.99	0.81	0.30	0.34	0.20	0.46	0.99	0.81	0.03	0.39
1efw	0.64	0.73	1.00	0.91	0.78	0.81	0.64	0.73	1.00	0.91	0.00	0.41
1ekz	0.19	0.19	0.89	0.96	0.31	0.31	0.30	0.25	0.89	0.96	0.17	0.09
1g1x	0.88	0.88	1.00	1.00	0.93	0.93	0.88	0.88	1.00	1.00	0.00	0.00
1hc8	0.94	0.95	1.00	1.00	0.97	0.97	0.94	0.94	1.00	1.00	0.00	0.07
1hvu	0.44	0.50	0.99	0.90	0.61	0.65	0.45	0.57	0.99	0.90	0.03	0.32
1jbr	0.92	0.92	1.00	1.00	0.96	0.96	0.92	0.92	1.00	1.00	0.00	0.00
1kog	0.15	0.16	1.00	0.94	0.27	0.28	0.15	0.25	1.00	0.94	0.00	0.12
1kq2	0.26	0.42	1.00	0.65	0.41	0.51	0.26	0.67	1.00	0.65	0.00	0.68
1m5o	0.47	0.66	1.00	0.91	0.64	0.77	0.47	0.74	1.00	0.91	0.00	0.57
1m8w	0.83	0.84	1.00	1.00	0.91	0.91	0.83	0.84	1.00	1.00	0.00	0.02
1mfq	0.78	0.78	1.00	1.00	0.88	0.87	0.78	0.78	1.00	1.00	0.02	0.00
1mms	0.89	0.90	1.00	0.99	0.94	0.94	0.89	0.89	1.00	0.99	0.00	0.13
1msw	0.08	0.91	1.00	0.99	0.16	0.95	0.08	0.99	1.00	0.99	0.00	0.99
1ob2	0.46	0.67	1.00	0.83	0.63	0.74	0.46	0.73	1.00	0.83	0.00	0.65
1t4l	0.79	0.79	1.00	1.00	0.88	0.88	0.79	0.79	1.00	1.00	0.00	0.00
1ttt	0.47	0.59	1.00	0.89	0.64	0.71	0.47	0.65	1.00	0.89	0.00	0.44
1u63	0.83	0.86	1.00	0.97	0.91	0.91	0.83	0.84	1.00	0.97	0.00	0.22
1wne	0.22	0.92	1.00	0.99	0.36	0.95	0.22	0.98	1.00	0.99	0.00	0.98
1zbi	0.22	0.37	1.00	0.71	0.37	0.49	0.22	0.67	1.00	0.71	0.00	0.66
2ad9	0.92	0.92	1.00	1.00	0.96	0.96	0.92	0.92	1.00	1.00	0.00	0.01
2adb	0.92	0.95	1.00	0.97	0.96	0.96	0.92	0.93	1.00	0.97	0.00	0.48
2adc	0.69	0.85	1.00	0.85	0.82	0.85	0.69	0.79	1.00	0.85	0.00	0.67
2b6g	0.52	0.52	1.00	1.00	0.69	0.69	0.52	0.52	1.00	1.00	0.00	0.00
2c0b	0.39	0.67	1.00	0.82	0.57	0.73	0.39	0.77	1.00	0.82	0.00	0.73
2dra	0.14	0.63	1.00	0.67	0.25	0.65	0.14	0.90	1.00	0.67	0.00	0.93
2err	0.83	0.92	1.00	0.97	0.90	0.95	0.83	0.91	1.00	0.97	0.00	0.60
2ez6	0.52	0.53	1.00	0.98	0.69	0.69	0.53	0.54	1.00	0.98	0.01	0.06
2hgh	0.54	0.76	1.00	0.80	0.70	0.78	0.54	0.75	1.00	0.80	0.00	0.70
2i91	0.17	0.81	1.00	0.85	0.29	0.83	0.17	0.94	1.00	0.85	0.00	0.96
2ix1	0.18	0.66	1.00	0.91	0.30	0.76	0.18	0.90	1.00	0.91	0.00	0.89
2py9	0.80	0.88	1.00	0.98	0.89	0.92	0.80	0.87	1.00	0.98	0.00	0.44
3bo2	0.37	0.55	1.00	0.89	0.54	0.68	0.37	0.69	1.00	0.89	0.00	0.57
3bsb	0.83	0.84	1.00	1.00	0.91	0.91	0.83	0.84	1.00	1.00	0.00	0.02
3bsx	0.84	0.85	1.00	1.00	0.92	0.92	0.84	0.85	1.00	1.00	0.00	0.02
3bx2	0.86	0.87	1.00	0.99	0.92	0.93	0.86	0.86	1.00	0.99	0.00	0.07

TABLE S3 – Résultats globaux des *Benchmarks* I et II : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

	Précision		Rappel		Fscore		*Accuracy*		Sensibilité		Spécificité	
pdb	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS	ROS	POS
1m5o	0.48	0.66	1.00	0.81	0.65	0.73	0.48	0.71	1.00	0.81	0.00	0.62
1qtq	0.28	0.73	1.00	0.76	0.44	0.74	0.28	0.85	1.00	0.76	0.00	0.89
1wpu	0.87	0.88	1.00	1.00	0.93	0.93	0.87	0.88	1.00	1.00	0.00	0.05
1yvp	0.94	0.94	1.00	1.00	0.97	0.97	0.94	0.94	1.00	1.00	0.00	0.00
1zbh	0.98	0.99	1.00	1.00	0.99	0.99	0.98	0.99	1.00	1.00	0.00	0.21
2ad9	0.00	0.00	1.00	1.00	0.00	0.01	0.02	0.98	1.00	1.00	0.02	0.98

TABLE S4 – Résultats globaux des 6 complexes avec protéine non liée : avec les mesures d'évaluation globales pour chacun des complexes pour la fonction de score par défaut de RosettaDock (ROS) et la fonction de score atomique apprise avec ROGER (POS).

pdb	ES		Top10			Top100		# presque-natifs	ROC-AUC	
	ROS	POS	ROS	POS	Attendu	ROS	POS		ROS	POS
1b7f	0.47	5.15	1	10	3.58	1	100	3579	0.32	0.89
1c9s	0.90	1.19	6	7	5.91	25	81	5905	0.37	0.69
1dk1	2.32	2.58	9	8	7.66	89	97	7660	0.61	0.82
1e7k	0.67	1.33	2	1	3.36	11	16	3359	0.50	0.53
1ec6	0.31	0.88	1	0	1.73	3	14	1730	0.44	0.60
1efw	0.41	2.90	0	10	6.37	0	96	6366	0.33	0.79
1ekz	0.95	0.81	4	1	1.77	18	20	1768	0.52	0.52
1g1x	3.32	0.54	2	6	8.76	85	82	8764	0.59	0.51
1hc8	1.56	3.63	10	10	9.43	94	100	9433	0.49	0.79
1hvu	0.77	1.93	2	2	4.37	42	48	4366	0.50	0.70
1jbr	3.56	0.02	9	9	9.23	71	86	9229	0.45	0.65
1kog	0.95	1.14	3	1	1.53	11	23	1532	0.49	0.54
1kq2	0.09	5.40	0	9	2.60	0	89	2602	0.31	0.74
1m5o	1.06	1.33	6	5	4.75	6	78	4748	0.24	0.83
1m8w	1.79	1.27	4	10	8.34	80	97	8343	0.37	0.74
1mfq	1.36	2.35	10	3	7.76	91	52	7757	0.62	0.59
1mms	2.52	3.31	10	9	8.86	85	96	8859	0.50	0.86
1msw	0.00	8.84	0	10	0.84	0	98	841	0.08	1.00
1ob2	0.78	3.38	7	10	4.63	12	97	4625	0.41	0.83
1t4l	0.56	1.64	10	10	7.85	98	95	7852	0.52	0.61
1ttt	0.76	3.25	6	10	4.69	11	97	4691	0.39	0.78
1u63	3.14	1.13	7	10	8.34	72	100	8338	0.44	0.86
1wne	0.01	4.41	0	7	2.21	0	96	2213	0.17	0.99
1zbi	0.37	2.95	0	8	2.23	0	69	2231	0.34	0.75
2ad9	1.99	2.53	10	10	9.18	84	100	9179	0.31	0.90
2adb	1.03	3.37	10	10	9.18	75	100	9180	0.25	0.89
2adc	3.38	2.61	3	10	6.94	40	100	6940	0.44	0.86
2b6g	1.73	0.02	9	6	5.23	93	62	5228	0.58	0.42
2c0b	0.00	2.98	0	8	3.94	0	91	3937	0.26	0.86
2dra	0.03	6.16	0	8	1.42	0	88	1419	0.32	0.91
2err	1.27	2.48	8	10	8.26	20	100	8263	0.15	0.92
2ez6	1.42	2.92	10	9	5.24	87	67	5235	0.58	0.62
2hgh	0.74	1.03	0	4	5.39	0	69	5387	0.25	0.82
2i91	0.47	7.35	0	10	1.73	0	95	1726	0.28	0.98
2ix1	0.02	2.14	0	4	1.80	0	55	1795	0.17	0.94
2py9	4.28	2.95	9	10	8.01	89	98	8014	0.50	0.83
3bo2	0.66	2.73	4	6	3.66	4	82	3663	0.31	0.80
3bsb	1.60	1.55	7	10	8.35	81	100	8348	0.36	0.80
3bsx	1.60	1.20	3	10	8.45	83	100	8448	0.32	0.82
3bx2	1.83	3.09	8	10	8.60	75	100	8595	0.33	0.84

TABLE S5 – Résultats pour les 40 complexes utilisés des *Benchmarks* I et II : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES		Top10			Top100		# presque-natifs	ROC-AUC	
	ROS	POS	ROS	POS	Attendu	ROS	POS		ROS	POS
1m5o	0.43	4.75	0	4	4.79	3	66	4790	0.25	0.79
1qtq	0.00	6.09	0	10	2.80	0	98	2798	0.24	0.91
1wpu	1.60	3.15	9	10	8.72	71	100	8723	0.47	0.70
1yvp	0.93	0.20	6	10	9.36	61	100	9362	0.29	0.81
1zbh	0.88	0.00	5	10	9.83	82	100	9834	0.12	0.96
2ad9	0.00	6.06	0	0	0.00	0	0	1	0.02	0.98

TABLE S6 – Résultats pour les 6 complexes dans le cas de la protéine non liée des *Benchmarks* I et II : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1asy	2.35	0	0.008	0	8	0.59
1av6	2.43	5	0.654	19	654	0.71
1b23	1.67	0	0.024	1	24	0.72
1c0a	1.89	0	0.001	0	1	0.77
1ddl	1.21	0	0.327	1	327	0.61
1dfu	1.20	2	0.743	9	743	0.66
1di2	0.93	0	0.114	0	114	0.52
1e8o	1.43	0	0.743	10	743	0.60
1f7u	2.56	0	0.007	1	7	0.86
1feu	1.97	2	0.378	17	378	0.69
1ffy	2.30	0	0.008	2	8	0.88
1fxl	4.43	8	0.589	64	589	0.85
1gtf	1.55	10	6.307	99	6307	0.70
1h3e	1.98	0	0.024	1	24	0.75
1h4s	1.73	0	0.036	1	36	0.67
1hq1	1.27	0	0.684	9	684	0.60
1j1u	1.66	0	0.112	1	112	0.54
1j2b	2.15	0	0.001	0	1	0.72
1jbs	2.57	0	0.499	20	499	0.71
1jid	1.91	1	0.377	11	377	0.65
1k8w	3.16	2	0.151	16	151	0.84
1knz	3.26	10	1.136	90	1136	0.73
1lng	1.80	2	0.13	6	130	0.65
1m8v	1.70	10	6.192	92	6192	0.61
1m8x	1.91	4	0.899	25	899	0.67
1mzp	1.85	2	0.148	8	148	0.69
1n35	4.40	2	0.017	17	17	1.00
1n78	2.08	0	0.01	1	10	0.83
1ooa	2.92	0	0.497	21	497	0.76
1pgl	1.85	2	0.898	26	898	0.63
1q2r	1.62	0	0.043	0	43	0.51
1qf6	2.06	0	0.005	1	5	0.85
1qtq	2.05	0	0.007	3	7	0.93
1r3e	1.78	0	0.028	0	28	0.59
1r9f	1.77	0	0.2	2	200	0.71
1sds	1.47	2	2.19	36	2190	0.67
1ser	1.95	0	0.008	1	8	0.88
1si3	2.33	4	0.547	23	547	0.69
1t0k	1.08	0	0.382	1	382	0.56
1tfw	2.99	0	0.004	0	4	0.98
1u0b	2.24	1	0.014	3	14	0.83
1un6	3.44	1	0.129	20	129	0.85
1uvj	5.06	8	0.357	62	357	0.94
1vfg	2.02	1	0.248	7	248	0.68
1wpu	1.53	1	1.162	21	1162	0.65
1wsu	1.26	0	2.288	20	2288	0.59
1wz2	2.18	0	0	0	0	1.00

Résultats de la fonction de score gros-grain VOR pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1yvp	1.72	1	0.13	4	130	0.64
1zbh	2.07	4	0.967	30	967	0.62
2a8v	2.22	10	5.018	90	5018	0.67
2anr	1.52	3	0.929	22	929	0.59
2asb	2.80	3	0.186	9	186	0.80
2az0	2.83	1	0.2	3	200	0.69
2azx	2.00	0	0.011	0	11	0.70
2b3j	2.39	2	0.426	29	426	0.71
2bgg	2.13	2	0.131	8	131	0.63
2bh2	3.23	0	0.051	12	51	0.84
2bte	2.46	0	0.002	1	2	0.98
2bu1	1.88	2	1.291	22	1291	0.65
2bx2	1.74	3	0.31	9	310	0.62
2ct8	1.92	0	0.029	1	29	0.64
2czj	2.69	2	0.207	9	207	0.76
2d6f	1.58	0	0.087	2	87	0.55
2der	1.10	0	0.029	0	29	0.72
2du3	1.20	0	0.186	1	186	0.56
2e9t	2.75	1	0.06	16	60	0.80
2f8k	1.33	3	3.82	52	3820	0.53
2f8s	1.21	0	0.076	0	76	0.39
2fk6	2.18	0	0.092	5	92	0.70
2fmt	1.82	0	0.018	0	18	0.67
2gic	4.48	7	0.103	38	103	0.94
2gje	2.11	0	0.079	3	79	0.72
2gjw	1.98	0	0.078	3	78	0.70
2gtt	3.91	5	0.069	24	69	0.91
2gxb	1.99	7	3.404	65	3404	0.64
2hw8	2.10	1	0.149	13	149	0.75
2i82	2.80	2	0.159	19	159	0.83
2iy5	1.22	0	0.001	0	1	0.51
2jlv	4.63	8	0.416	59	416	0.90
2nqp	2.10	1	0.296	15	296	0.70
2nug	2.19	0	0.001	0	1	0.97
2ozb	2.20	2	0.078	4	78	0.74
2pjp	2.14	1	0.576	15	576	0.65
2po1	1.91	1	0.19	4	190	0.65
2qux	2.53	1	0.26	7	260	0.74
2r7r	3.19	3	0.074	25	74	0.95
2r8s	1.31	0	0.039	0	39	0.56
2vnu	5.79	8	0.123	38	123	0.98
2voo	4.88	10	3.261	97	3261	0.77
2w2h	1.30	0	0.018	0	18	0.46
2wj8	3.08	3	0.43	29	430	0.79
2z2q	1.73	3	0.863	20	863	0.57
2zi0	0.86	3	1.278	15	1278	0.49
2zko	3.61	1	0.181	11	181	0.80

Résultats de la fonction de score gros-grain VOR pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
2zni	1.91	1	0.017	2	17	0.83
2zue	2.77	0	0.004	0	4	0.95
2zzm	3.59	2	0.009	4	9	0.93
3a6p	2.19	0	0.001	1	1	0.99
3bso	2.74	2	0.038	12	38	0.85
3bt7	3.59	6	0.192	36	192	0.87
3ciy	1.43	0	0.01	2	10	0.79
3d2s	1.63	10	5.557	93	5557	0.62
3dd2	1.95	1	0.232	6	232	0.74
3egz	1.41	0	0.235	5	235	0.57
3eph	2.01	0	0.001	0	1	0.99
3eqt	3.87	2	0.405	24	405	0.80
3ex7	2.03	1	0.239	12	239	0.71
3fht	2.05	2	0.513	8	513	0.65
3foz	1.81	0	0.019	4	19	0.89
3gib	1.67	1	0.641	13	641	0.63
3hax	2.07	0	0.008	0	8	0.62
3hl2	2.07	0	0.015	1	15	0.67
3htx	1.59	0	0.014	0	14	0.65
3i5x	2.48	0	0.175	3	175	0.74
3iab	3.35	0	0.03	3	30	0.82
3icq	2.64	0	0.002	0	2	0.91
3iev	1.90	1	0.227	8	227	0.68
3k62	2.34	4	0.486	19	486	0.69
3l25	7.94	3	0.247	26	247	0.97
3snp	2.68	0	0.041	8	41	0.85

TABLE S7 – Résultats de la fonction de score gros-grain VOR pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1asy	0.43	0	0.008	0	8	0.35
1av6	0.46	0	0.654	0	654	0.37
1b23	1.16	0	0.024	0	24	0.60
1c0a	0.79	0	0.001	0	1	0.04
1ddl	1.12	0	0.327	0	327	0.53
1dfu	0.31	1	0.743	2	743	0.31
1di2	0.82	0	0.114	1	114	0.39
1e8o	0.88	0	0.743	0	743	0.54
1f7u	0.38	0	0.007	0	7	0.16
1feu	0.23	0	0.378	1	378	0.38
1ffy	0.65	0	0.008	0	8	0.15
1fxl	0.46	0	0.589	2	589	0.38
1gtf	1.27	7	6.307	53	6307	0.48
1h3e	0.51	0	0.024	0	24	0.36
1h4s	0.73	0	0.036	0	36	0.34
1hq1	0.50	0	0.684	0	684	0.41
1j1u	0.65	0	0.112	0	112	0.43
1j2b	0.16	0	0.001	0	1	0.12
1jbs	0.92	0	0.499	2	499	0.56
1jid	0.38	0	0.377	0	377	0.32
1k8w	1.58	0	0.151	0	151	0.47
1knz	0.10	0	1.136	1	1136	0.26
1lng	0.32	0	0.13	0	130	0.34
1m8v	0.48	0	6.192	34	6192	0.44
1m8x	0.40	0	0.899	6	899	0.41
1mzp	0.60	0	0.148	0	148	0.40
1n35	0.12	0	0.017	0	17	0.01
1n78	0.46	0	0.01	0	10	0.13
1ooa	0.15	0	0.497	2	497	0.27
1pgl	0.35	0	0.898	0	898	0.30
1q2r	1.00	0	0.043	1	43	0.46
1qf6	0.81	0	0.005	0	5	0.18
1qtq	0.59	0	0.007	0	7	0.09
1r3e	0.73	0	0.028	0	28	0.39
1r9f	0.49	0	0.2	0	200	0.38
1sds	0.69	4	2.19	15	2190	0.52
1ser	0.33	0	0.008	0	8	0.12
1si3	0.18	0	0.547	0	547	0.25
1t0k	0.19	0	0.382	1	382	0.33
1tfw	0.27	0	0.004	0	4	0.13
1u0b	0.37	0	0.014	0	14	0.14
1un6	0.21	0	0.129	0	129	0.20
1uvj	0.02	0	0.357	0	357	0.09
1vfg	0.51	0	0.248	0	248	0.40
1wpu	0.62	0	1.162	5	1162	0.48
1wsu	0.71	0	2.288	7	2288	0.50
1wz2	0.29	0	0	0	0	1.00

Résultats de la fonction de score gros-grain avec poids positifs pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1yvp	0.89	0	0.13	1	130	0.33
1zbh	0.23	0	0.967	0	967	0.29
2a8v	0.69	5	5.018	54	5018	0.46
2anr	0.35	0	0.929	1	929	0.38
2asb	0.32	0	0.186	0	186	0.27
2az0	0.64	0	0.2	0	200	0.32
2azx	0.76	0	0.011	0	11	0.33
2b3j	0.34	0	0.426	0	426	0.28
2bgg	0.52	0	0.131	0	131	0.50
2bh2	0.84	0	0.051	0	51	0.40
2bte	0.33	0	0.002	0	2	0.02
2bu1	0.35	1	1.291	4	1291	0.38
2bx2	0.90	0	0.31	0	310	0.35
2ct8	0.84	0	0.029	0	29	0.39
2czj	0.63	0	0.207	0	207	0.34
2d6f	0.53	0	0.087	0	87	0.49
2der	1.27	0	0.029	0	29	0.45
2du3	0.78	0	0.186	2	186	0.44
2e9t	0.40	0	0.06	0	60	0.27
2f8k	0.66	0	3.82	14	3820	0.47
2f8s	0.87	0	0.076	3	76	0.62
2fk6	0.48	0	0.092	0	92	0.34
2fmt	0.15	0	0.018	0	18	0.25
2gic	0.18	0	0.103	0	103	0.07
2gje	0.26	0	0.079	0	79	0.28
2gjw	0.45	0	0.078	0	78	0.54
2gtt	0.01	0	0.069	0	69	0.07
2gxb	0.16	1	3.404	8	3404	0.32
2hw8	0.27	0	0.149	0	149	0.21
2i82	0.49	0	0.159	0	159	0.21
2iy5	0.47	0	0.001	0	1	0.51
2jlv	0.54	0	0.416	1	416	0.17
2nqp	0.27	0	0.296	0	296	0.34
2nug	0.34	0	0.001	0	1	0.28
2ozb	1.10	0	0.078	2	78	0.52
2pjp	0.70	0	0.576	1	576	0.40
2po1	0.16	0	0.19	0	190	0.35
2qux	0.06	0	0.26	0	260	0.20
2r7r	0.48	0	0.074	0	74	0.45
2r8s	0.62	0	0.039	0	39	0.50
2vnu	0.21	0	0.123	0	123	0.18
2voo	0.00	0	3.261	3	3261	0.34
2w2h	0.81	0	0.018	0	18	0.52
2wj8	0.01	0	0.43	0	430	0.17
2z2q	0.71	0	0.863	2	863	0.45
2zi0	1.07	1	1.278	14	1278	0.51
2zko	0.32	0	0.181	0	181	0.17

Résultats de la fonction de score gros-grain avec poids positifs pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
2zni	0.45	0	0.017	0	17	0.16
2zue	0.18	0	0.004	0	4	0.15
2zzm	0.37	0	0.009	0	9	0.11
3a6p	0.34	0	0.001	0	1	0.31
3bso	0.57	0	0.038	0	38	0.22
3bt7	0.63	0	0.192	0	192	0.17
3ciy	0.89	0	0.01	0	10	0.23
3d2s	1.00	5	5.557	47	5557	0.49
3dd2	0.68	0	0.232	0	232	0.36
3egz	0.67	0	0.235	1	235	0.38
3eph	0.18	0	0.001	0	1	0.01
3eqt	0.21	0	0.405	0	405	0.22
3ex7	1.09	0	0.239	1	239	0.46
3fht	0.88	0	0.513	2	513	0.51
3foz	0.82	0	0.019	0	19	0.20
3gib	0.62	0	0.641	2	641	0.41
3hax	0.83	0	0.008	0	8	0.40
3hl2	0.48	0	0.015	0	15	0.42
3htx	0.77	0	0.014	0	14	0.36
3i5x	0.19	0	0.175	0	175	0.21
3iab	0.05	0	0.03	0	30	0.15
3icq	0.20	0	0.002	0	2	0.08
3iev	0.31	0	0.227	0	227	0.19
3k62	0.34	0	0.486	1	486	0.31
3l25	0.04	0	0.247	0	247	0.08
3snp	0.63	0	0.041	0	41	0.31

TABLE S8 – Résultats de la fonction de score gros-grain avec poids positifs pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1asy	0.69	0	0.008	0	8	0.46
1av6	0.93	0	0.654	2	654	0.48
1b23	0.77	0	0.024	0	24	0.48
1c0a	0.81	0	0.001	0	1	0.48
1ddl	1.04	0	0.327	1	327	0.48
1dfu	1.08	2	0.743	14	743	0.52
1di2	1.06	1	0.114	1	114	0.45
1e8o	0.80	0	0.743	5	743	0.49
1f7u	0.52	0	0.007	0	7	0.38
1feu	1.07	1	0.378	4	378	0.48
1ffy	0.65	0	0.008	0	8	0.46
1fxl	0.63	0	0.589	0	589	0.45
1gtf	1.12	8	6.307	65	6307	0.51
1h3e	0.80	0	0.024	0	24	0.39
1h4s	0.86	0	0.036	0	36	0.46
1hq1	0.96	1	0.684	7	684	0.50
1j1u	0.98	1	0.112	3	112	0.55
1j2b	0.73	0	0.001	0	1	0.51
1jbs	0.86	0	0.499	1	499	0.49
1jid	1.04	0	0.377	4	377	0.48
1k8w	0.71	0	0.151	0	151	0.47
1knz	0.81	1	1.136	11	1136	0.50
1lng	0.97	0	0.13	0	130	0.45
1m8v	1.07	5	6.192	57	6192	0.49
1m8x	1.04	1	0.899	7	899	0.51
1mzp	0.93	0	0.148	0	148	0.44
1n35	0.73	0	0.017	0	17	0.27
1n78	0.71	0	0.01	0	10	0.45
1ooa	0.74	1	0.497	3	497	0.46
1pgl	1.04	2	0.898	6	898	0.50
1q2r	1.07	0	0.043	0	43	0.53
1qf6	0.85	0	0.005	0	5	0.52
1qtq	0.69	0	0.007	0	7	0.34
1r3e	1.07	0	0.028	0	28	0.45
1r9f	0.98	0	0.2	2	200	0.43
1sds	1.04	1	2.19	16	2190	0.50
1ser	0.82	0	0.008	0	8	0.41
1si3	0.80	0	0.547	3	547	0.46
1t0k	0.88	0	0.382	3	382	0.50
1tfw	0.95	0	0.004	0	4	0.61
1u0b	0.71	0	0.014	1	14	0.55
1un6	0.34	0	0.129	0	129	0.36
1uvj	0.94	1	0.357	5	357	0.51
1vfg	0.81	0	0.248	1	248	0.44
1wpu	0.96	2	1.162	6	1162	0.49
1wsu	1.02	4	2.288	29	2288	0.51
1wz2	0.60	0	0	0	0	1.00

Résultats de la fonction de score gros-grain non linéaire avec valeurs de centrage pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
1yvp	0.99	0	0.13	3	130	0.48
1zbh	0.87	0	0.967	4	967	0.48
2a8v	0.77	3	5.018	43	5018	0.49
2anr	0.71	0	0.929	4	929	0.47
2asb	0.86	0	0.186	0	186	0.46
2az0	1.40	1	0.2	5	200	0.51
2azx	0.83	0	0.011	1	11	0.52
2b3j	0.66	0	0.426	1	426	0.49
2bgg	0.92	0	0.131	2	131	0.53
2bh2	0.65	0	0.051	1	51	0.39
2bte	0.55	0	0.002	0	2	0.17
2bu1	0.96	1	1.291	8	1291	0.49
2bx2	1.09	0	0.31	0	310	0.48
2ct8	0.75	0	0.029	0	29	0.39
2czj	0.75	0	0.207	1	207	0.49
2d6f	1.00	0	0.087	3	87	0.53
2der	0.94	0	0.029	0	29	0.48
2du3	0.91	0	0.186	3	186	0.47
2e9t	0.91	1	0.06	2	60	0.49
2f8k	1.06	5	3.82	43	3820	0.49
2f8s	1.04	0	0.076	2	76	0.53
2fk6	0.65	0	0.092	0	92	0.41
2fmt	0.89	0	0.018	0	18	0.41
2gic	0.40	0	0.103	0	103	0.42
2gje	0.69	0	0.079	0	79	0.49
2gjw	0.89	0	0.078	0	78	0.47
2gtt	0.40	0	0.069	0	69	0.34
2gxb	0.96	4	3.404	39	3404	0.49
2hw8	0.59	0	0.149	0	149	0.39
2i82	0.86	0	0.159	2	159	0.46
2iy5	0.87	0	0.001	0	1	0.02
2jlv	1.01	1	0.416	6	416	0.50
2nqp	0.84	0	0.296	1	296	0.51
2nug	1.09	0	0.001	0	1	0.42
2ozb	0.64	0	0.078	0	78	0.47
2pjp	0.86	0	0.576	1	576	0.49
2po1	0.86	0	0.19	1	190	0.48
2qux	0.79	0	0.26	1	260	0.46
2r7r	0.73	0	0.074	0	74	0.40
2r8s	0.85	0	0.039	0	39	0.52
2vnu	0.49	0	0.123	0	123	0.27
2voo	1.07	3	3.261	27	3261	0.49
2w2h	1.14	0	0.018	0	18	0.47
2wj8	0.74	0	0.43	2	430	0.47
2z2q	0.97	2	0.863	10	863	0.49
2zi0	1.09	1	1.278	13	1278	0.51
2zko	1.37	1	0.181	6	181	0.53

Résultats de la fonction de score gros-grain non linéaire avec valeurs de centrage pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	ES	Top10	Attendu	Top100	# presque-natifs	ROC-AUC
2zni	0.75	0	0.017	0	17	0.47
2zue	0.64	0	0.004	0	4	0.26
2zzm	0.67	0	0.009	0	9	0.36
3a6p	1.16	0	0.001	0	1	0.11
3bso	0.66	0	0.038	0	38	0.38
3bt7	0.47	0	0.192	0	192	0.45
3ciy	1.07	0	0.01	0	10	0.62
3d2s	0.91	3	5.557	49	5557	0.48
3dd2	0.87	1	0.232	6	232	0.50
3egz	0.98	0	0.235	0	235	0.47
3eph	0.83	0	0.001	0	1	0.59
3eqt	0.40	0	0.405	1	405	0.45
3ex7	0.85	0	0.239	1	239	0.47
3fht	0.99	0	0.513	3	513	0.49
3foz	0.72	0	0.019	0	19	0.30
3gib	0.84	0	0.641	3	641	0.49
3hax	0.86	0	0.008	0	8	0.39
3hl2	0.76	0	0.015	0	15	0.51
3htx	0.99	0	0.014	0	14	0.54
3i5x	0.90	0	0.175	1	175	0.52
3iab	0.67	0	0.03	0	30	0.52
3icq	0.71	0	0.002	0	2	0.46
3iev	0.82	0	0.227	0	227	0.45
3k62	0.87	0	0.486	3	486	0.48
3l25	0.35	0	0.247	0	247	0.42
3snp	0.70	0	0.041	0	41	0.50

TABLE S9 – Résultats de la fonction de score gros-grain non linéaire avec valeurs de centrage pour les 120 complexes de la PRIDB : score d'enrichissement (ES), nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100, nombre de presque-natifs sur les 10 000 structures et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

pdb	Description protéine	Nombre d'acides aminés	Description ARN	Nombre d'acides nucléiques	Type
2zi0	Tomato aspermy virus protein 2b	60	siRNA	40	dsRNA
2gxb	H. sapiens dsRNA-specific adenosine deaminase	62	dsRNA 5'-UCGCGCG-3' and 5'-CGCGCG-3'	13	dsRNA
1hq1	E. coli signal recognition particle protein	76	4.5S RNA domain IV	49	dsRNA
2f8k	S. cerevisiae VTS1 protein SAM domain	84	5'-UAAUCUUUGACAGAUU-3'	16	dsRNA
1lng	M. jannaschii signal recognition particle 19 kDa protein	87	7S.S signal recognition particle RNA	97	dsRNA
3egz	H. sapiens U1 small nuclear ribonucleoprotein A	91	Tetracyclin haptamer and artificial riboswitch	65	dsRNA
1dfu	E. coli ribosomal protein L25	94	5S rRNA fragment	38	dsRNA
1jid	H. sapiens signal recognition particle 19 kDa protein	114	signal recognition particle RNA helix 6	29	dsRNA
2pjp	E. coli selenocysteine-specific elongation factor selB	121	SECIS RNA	23	dsRNA
1r9f	Tomato bushy stunt virus core protein 19	121	siRNA	40	dsRNA
2czj	T. thermophilus SsrA-binding protein	122	tmRNA tRNA domain	62	dsRNA
1wsu	M. thermoacetica selenocystein-specific elongation factor	124	5'-GGCGUUGCCGGUCGGCAACGCC-3'	22	dsRNA
2bu1	E. phage MS2 coat protein	129	5'-CAUGAGGAUUACCCAUG-3'	17	dsRNA
1di2	X. laevis protein A dsRNA-binding domain	129	dsRNA 5'-GGCGCGCGCC-3' tetramer	40	dsRNA
2zko	Influenza A virus non-structural protein 1 (NS1)	140	dsRNA	42	dsRNA
2az0	Flock house virus B2 protein	141	dsRNA 5'-GCAUGGACGCGUCCAUGC-3' dimer	36	dsRNA
1jbs	A. restrictus restrictocin	142	29-mer sarcin/ricin domain RNA analog	29	dsRNA
1un6	X. laevis transcription factor IIIA	145	5S rRNA fragment	61	dsRNA
1e8o	H. sapiens signal recognition particle 9 kDa and 14 kDa protein	149	7SL RNA	49	dsRNA
2anr	H. sapiens neuro-oncological ventral antigen 1 (NOVA 1)	155	5'-CUCGCGGAUCAGUCACCCAAGCGAG-3'	25	dsRNA
1feu	T. thermophilus 50S ribosomal protein L25	185	5S rRNA fragment	40	dsRNA
2i82	E. coli ribosomal large subunit pseudouridine synthase A	217	5'-GAGGGGAAUGAAAAUCCCCUC-3'	21	dsRNA
1mzp	S. acidocaldarius 50S ribosomal protein L1P	217	23S rRNA fragment	55	dsRNA
1zbh	H. sapiens 3'-5' exonuclease ERI1	224	histone mRNA stem-loop	16	dsRNA
2hw8	T. thermophilus 50S ribosomal protein L1	228	mRNA	36	dsRNA
2qux	P. phage coat protein	244	dsRNA	25	dsRNA
3iab	S. cerevisiae ribonuclease P/MRP protein subunit POP6 and POP7	255	Ribonuclease MRP RNA component P3 domain	46	dsRNA
3eqt	H. sapiens ATP-dependent RNA helicase DHX58	269	5'-GCGCGCGC-3'	8	dsRNA
3dd2	H. sapiens thrombine	288	dsRNA	26	dsRNA
2b3j	S. aureus tRNA adenosine deaminase (tAdA)	302	tRNA-ARG2 anticodon stem-loop	15	dsRNA
1k8w	E. coli tRNA-PseudoU synthase B	304	T stem-loop RNA	22	dsRNA

Codes PDB des complexes protéine-ARN utilisés dans la PRIDB non redondante RB199, avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés #aa et d'acides nucléiques #an dans la structure 3D. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

pdb	Description protéine	Nombre d'acides aminés	Description ARN	Nombre d'acides nucléiques	Type
1r3e	T. maritima tRNA-PseudoU synthase B	305	tRNA-PseudoU T-arm	51	dsRNA
1ooa	M. musculus nuclear factor NF-kappa-B p105 subunit	313	RNA aptamer	29	dsRNA
1vfg	A. aeolicus tRNA nucleotidyltransferase	342	Primer tRNA	31	dsRNA
2ozb	H. sapiens U4/U6 small nuclear ribonucleoprotein Prp31	365	U4snRNA 5' stem-loop	33	dsRNA
3bt7	E. coli tRNA (uracile-5)-methyltransferase	369	19 nucleotide T-arm analogue	19	dsRNA
2bgg	A. fulgidus piwi AF1318 protein	395	dsRNA 5'-UUCGACGC-3' and 5'-GUCGAAUU-3'	16	dsRNA
2r8s	M. musculus synthetic FAB	433	P4-P6 RNA ribozyme domain	159	dsRNA
2nug	A. aeolicus ribonuclease III	434	dsRNA	44	dsRNA
1t0k	E. coli maltose-binding periplasmic-protein and S. cerevisiae 60S ribosomal protein L30	462	mRNA 3' consensus 5'-UGACC-3'	27	dsRNA
2e9t	Foot-and-mouth disease virus RNA-dependent RNA polymerase	474	dsRNA 5'-UAGGGCCC-3' and 5'-GGGCCCU-3'	15	dsRNA
3bso	Norwalk virus RNA-dependent RNA polymerase	479	primer-template RNA	16	dsRNA
3l25	Ebola virus polymerase cofactor VP35	492	dsRNA 5'-CGCAUGCG-3' dimer	16	dsRNA
1yvp	X. laevis 60 kDa SS-A Ro ribonucleoprotein	529	Both strands of the Y RNA sequence	20	dsRNA
2w2h	Equine infectious anemia virus protein TAT and E. caballus cyclin-T1	581	TAR RNA	44	dsRNA
2gjw	A. fulgidus tRNA-splicing endonuclease	612	dsRNA	37	dsRNA
3ciy	M. musculus toll-like receptor 3	661	dsRNA	92	dsRNA
1q2r	Z. mobilis queunine tRNA ribosyltransferase	748	dsRNA 5'-AGCACGGCUNUAAACCGUGC-3' dimer	40	dsRNA
3htx	A. thaliana RNA methyltransferase HEN1	780	dsRNA	44	dsRNA
3snp	O. cuniculus cytoplasmic aconitate hydratase	850	Ferritin H IRE RNA	30	dsRNA
1tfw	A. fulgidus tRNA nucleotidyltransferase	874	Immature tRNA	26	dsRNA
3icq	S. cerevisiae GTP-binding nuclear protein GSP1/CNR1 and S. pombe exportin-T	1116	dsRNA	62	dsRNA
3a6p	H. sapiens exportin-5	1242	pre-miRNA	46	dsRNA
1n35	M. orthoreovirus minor core protein lambda 3	1264	dsRNA 5'-GGGGG-3' and 5'-UAGCCCCC-3'	13	dsRNA
2f8s	A. aeolicus argonaute protein	1408	dsRNA 5'-AGACAGCAUAUAUGCUGUCUUU-3' dimer	44	dsRNA
1m8v	P. abyssi putative snRNP sm-like protein	72	5'-UUUUUU-3'	6	ssRNA
1sds	M. jannaschii 50S ribosomal protein L7Ae	112	Archeal box H/ACA sRNA K-turn	15	ssRNA
2a8v	E. coli rho-dependent transcription termination factor RNA-binding domain	118	5'-CCC-3'	3	ssRNA
1si3	H. sapiens eukaryotic translation initiation factor 2C 1	120	5'-CGUGACUCU-3'	9	ssRNA

Codes PDB des complexes protéine-ARN utilisés dans la PRIDB non redondante RB199, avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés #aa et d'acides nucléiques #an dans la structure 3D. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

pdb	Description protéine	Nombre d'acides aminés	Description ARN	Nombre d'acides nucléiques	Type
3d2s	H. sapiens musculeblind-like protein 1 (MBNL1)	135	5'-GCUGU-3'	5	ssRNA
1gtf	G. staerothermophilus TRP RNA-binding attenuation protein	142	5'-GAGU-3'	4	ssRNA
1wpu	B. subtilis hut operon positive regulatory protein	148	5'-UUGAGUU-3'	7	ssRNA
1fxl	P. encephalomyelitis antigen hud	167	5'-UUUUAUUUU-3'	9	ssRNA
2voo	H. sapiens lupus LA protein	179	5'-UUU-3'	3	ssRNA
3gib	E. coli protein Hfq	191	5'-AAAAAAAAA-3'	9	ssRNA
2asb	M. tuberculosis transcription elongation protein nusA	226	5'-GAACUCAAUAG-3'	11	ssRNA
1av6	Vaccinia cap-specific mRNA methyltransferase VP39	290	M7G Capped RNA 5'-GAAAAA-3'	7	ssRNA
1knz	S. rotavirus nonstructural RNA-binding protein 34	292	mRNA 3' consensus 5'-UGACC-3'	5	ssRNA
2gje	T. brucei mRNA-binding protein bBP	292	Guide-RNA fragment	18	ssRNA
3iev	A. aeolicus GTP-binding protein ERA	302	16S rRNA 3' end	11	ssRNA
1m8x	H. sapiens pumilio 1 homology domain	341	5'-UUGUAUAU-3'	8	ssRNA
2wj8	Human respiratory syncytial virus nucleoprotein	374	5'-CCCCCCC-3'	7	ssRNA
3fht	H. sapiens ATP-dependent RNA helicase DDX19B	392	5'-UUUUUU-3'	6	ssRNA
3k62	C. elegans Fem-3 mRNA-binding factor 2	400	5'-UGUGUUAUC-3'	9	ssRNA
2gtt	Rabies virus nucleoprotein	401	ssRNA	17	ssRNA
2gic	Vesicular stomatitis Indiana virus nucleocapsid protein	416	ssRNA fragment	15	ssRNA
2bh2	E. coli 23S ribosomal RNA (uracil-5)-methyltransferase RUMA	419	23S rRNA fragment	29	ssRNA
2jlv	Dengue virus serine protease subunit NS3	451	5'-AGACUAA-3'	7	ssRNA
2bx2	E. coli ribonuclease E	500	5'-UUUACAGUAUUUGU-3'	14	ssRNA
2po1	P. abyssi probable exosome complex exonuclease	503	5'-AAAAAAA-3'	7	ssRNA
3i5x	S. cerevisiae ATP-dependent RNA helicase MSS116	509	5'-UUUUUUUUUU-3'	10	ssRNA
1ddl	Desmodium yellow mottle tymovirus	551	5'-UUUUUUU-3'	7	ssRNA
1pgl	Bean pod mottle virus	555	5'-AGUCUC-3'	6	ssRNA
1uvj	P. phage P2 protein	664	5'-UUCC-3'	4	ssRNA
3ex7	H. sapiens CASC3 and mago nashi homolog and RNA-binding protein 8 and eukaryotic initiation factor 4A-III	687	5'-UUUUUU-3'	6	ssRNA
2vnu	S. cerevisiae exosome complex exonuclease RRP44	694	5'-AAAAAAAAA-3'	9	ssRNA
2z2q	Flock house viruse capsid protein	980	Flock house virus genomic RNA	12	ssRNA
2r7r	Simian rotavirus RNA-dependent RNA polymerase	1073	5'-UGUGACC-3'	7	ssRNA
1j1u	M. jannaschii tRNA-TYR synthetase	300	tRNA-TYR	74	tRNA
3foz	E. coli isopentenyl-tRNA transferase	303	tRNA-PHE	69	tRNA

Codes PDB des complexes protéine-ARN utilisés dans la PRIDB non redondante RB199, avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés #aa et d'acides nucléiques #an dans la structure 3D. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

pdb	Description protéine	Nombre d'acides aminés	Description ARN	Nombre d'acides nucléiques	Type
2fk6	B. subtilis ribonuclease Z	307	tRNA-THR	53	tRNA
2fmt	E. coli tRNA-FMET formyltransferase	314	Formyl-methionyl-tRNA-FMET2	77	tRNA
2zzm	M. jannaschii uncharacterized protein MJ0883	329	tRNA-LEU	84	tRNA
2der	E. coli tRNA-specific 2-thiouridylase mnmA	348	tRNA-GLU	74	tRNA
3eph	S. cerevisiae tRNA isopentenyltransferase	402	tRNA	69	tRNA
1b23	T. aquaticus elongation factor EF-TU	405	E. coli tRNA-CYS	74	tRNA
1h3e	T. thermophilus tRNA-TYR synthetase	427	Wild type tRNA-TYR (GUA)	84	tRNA
1u0b	E. coli tRNA-CYS	461	tRNA-CYS	74	tRNA
2ct8	A. aeolicus tRNA-MET synthetase	465	tRNA-MET	74	tRNA
1n78	T. thermophilus tRNA-GLU synthetase	468	tRNA-GLU	75	tRNA
2d6f	M. thermautotrophicus tRNA-GLN amidotransferase	485	tRNA-GLN	72	tRNA
1asy	Yeast tRNA-ASP synthetase	490	tRNA-ASP	75	tRNA
3hax	P. furiosus probable tRNA-PseudoU synthase B and ribosome biogenesis protein Nop10 and 50S ribosomal protein L7Ae	503	H/ACA RNA	74	tRNA
2nqp	E. coli tRNA-PseudoU synthase A	528	tRNA-LEU	71	tRNA
1qtq	E. coli tRNA-GLN synthetase	529	tRNA-GLN II	74	tRNA
2zni	D. hafniense tRNA-PYL synthetase	558	tRNA-PYL	72	tRNA
1c0a	E. coli tRNA-ASP synthetase	585	tRNA-ASP	77	tRNA
1f7u	S. cerevisiae tRNA-ARG synthetase	607	tRNA-ARG	76	tRNA
2zue	P. horikoshii tRNA-ARG synthetase	628	tRNA-ARG	76	tRNA
1qf6	E. coli tRNA-THR synthetase	641	tRNA-THR	76	tRNA
2azx	H. sapiens tRNA-TRP synthetase	778	tRNA-TRP	72	tRNA
1ser	T. thermophilus tRNA-SER synthetase	793	tRNA-SER	65	tRNA
2bte	T. thermophilus aminoacyl-tRNA synthetase	877	tRNA-LEU transcript with anticodon CAG	78	tRNA
3hl2	H. sapiens O-phosphoseryl-tRNA-SEC selenium transferase	886	tRNA-SEC	82	tRNA
1ffy	S. aureus tRNA-ILE synthetase	917	tRNA-ILE	75	tRNA
1h4s	T. thermophilus tRNA-PRO synthetase	946	tRNA-PRO (CGG)	67	tRNA
1wz2	P. horikoshii tRNA-LEU synthetase	948	tRNA-LEU	88	tRNA
2du3	A. fulgidus O-phosphoseryl-tRNA synthetase	1001	tRNA-CYS	71	tRNA
2iy5	T. thermophilus tRNA-PHE synthetase	1117	tRNA-PHE	76	tRNA
1j2b	P. horikoshii archaeosine tRNA-GUA transglycosylase	1152	tRNA-VAL	77	tRNA

TABLE S10 – Codes PDB des complexes protéine-ARN utilisés dans la PRIDB non redondante RB199, avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés et d'acides nucléiques dans la structure 3D. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

D	pdb	pdb aa	Description protéine	Nombre d'acides aminés	pdb na	Description ARN	Nombre d'acides nucléiques	Type
R	2b6g	2d3d	S. cerevisiae VTS1 protein SAM domain	81	2b7g	5'-GGAGGCUCUGGCAGCUUUC-3'	19	dsRNA
R	1ec6	1dtj	H. sapiens neuro-oncological ventral antigen 2 (NOVA 2)	87	–	dsRNA	20	dsRNA
R	1t4l	1t4o	S. cerevisiae ribonuclease III	90	–	snRNA 47 precursor 5' terminal hairpin	32	dsRNA
R	1m5o	1nu4	H. sapiens U1 small nuclear ribonucleoprotein A	92	–	RNA substrate and RNA hairpin ribozyme	113	dsRNA
R	3bo2	1nu4	H. sapiens U1 small nuclear ribonucleoprotein A	95	–	Group I intron P9 and mRNA	221	dsRNA
R	1g1x	1ris	T. thermophilus 30S ribosomal protein S6	98	–	16S rRNA fragment	84	dsRNA
R	1zbi	1zbf	B. halodurans Ribonuclease H-related protein	135	–	RNA-DNA hydrid	24	dsRNA
R	1jbr	1aqz	A. restrictus restrictocin	149	–	31-mer SRD RNA analog	31	dsRNA
R	1u63	1l2a	M. jannaschii 50S ribosomal protein L1P	214	–	mRNA	49	dsRNA
R	2ez6	1jfz	A. aeolicus ribonuclease III	218	–	dsRNA	56	dsRNA
R	1kog	1evl	E. coli tRNA-THR synthetase	401	–	tRNA-THR synthetase mRNA operator essential domain	37	dsRNA
R	2dra	1r89	A. fulgidus CCA-adding enzyme	437	1vfg	tRNA mini DCC	34	dsRNA
R	1wne	1u09	Foot-and-mouth disease virus RNA-dependent RNA polymerase	476	–	dsRNA 5'-CAUGGGCC-3' and 5'-GGCCC-3' dimer	13	dsRNA
R	1c9s	1qaw	G. staerothermopilus TRP RNA-binding attenuation protein	67	–	5'-GAUGAGA-3'	7	ssRNA
R	1m8w	1mz8	H. sapiens pumilio 1 homology domain	340	–	5'-UUGUAUAU-3'	8	ssRNA
R	3bsb	1m8z	H. sapiens pumilio 1 homology domain	341	–	5'-UUUAAUGUU-3'	9	ssRNA
R	3bsx	1m8z	H. sapiens pumilio 1 homology domain	341	–	5'-UUGUAAUAUU-3'	10	ssRNA
R	1kq2	1kq1	S. aureus host factor protein for Q beta	365	–	5'-AUUUUUG-3'	7	ssRNA
R	2i91	1yvr	S. laevis 60 kDa SS-A Ro ribonucleoprotein	526	–	misfolded RNA fragment	23	ssRNA

Codes PDB des 40 complexes protéine-ARN utilisés dans les *benchmarks* I et II, par difficulté (D, avec R pour corps rigide, S pour semi-flexible et X pour flexible), avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés #aa et d'acides nucléiques #an dans la structure 3D et le code PDB de la structure 3D de la protéine (pdb aa) et de l'ARN (pdb na) non liés si disponibles. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

D	pdb	pdb aa	Description protéine	Nombre d'acides aminés	pdb na	Description ARN	Nombre d'acides nucléiques	Type
R	2ix1	2id0	E. coli exoribonuclease II	643	–	5'-AAAAAAAAAAAAA-3'	13	ssRNA
R	1ttt	1eft	T. aquaticus elongation factor EF-TU	405	4tna	Yeast tRNA-PHE	76	tRNA
R	1efw	1l0w	T. thermophilus tRNA-ASP synthetase	580	1c0a	tRNA-ASP	73	tRNA
S	1hc8	1foy	G. staerothermopilus ribosomal protein L11	74	–	23S rRNA fragment	57	dsRNA
S	1dk1	2fkx	T. thermophilus 30S ribosomal protein S15	86	–	30S rRNA fragment	57	dsRNA
S	1mfq	1qb2	H. sapiens signal recognition particle 54 kDa protein	108	1l9a	7S RNA of Human signal recognition particle	128	dsRNA
S	1e7k	2jnb	H. sapiens 15.5 kDa RNA-binding protein	125	–	5'-GCCAAUGAGGCCGAGGC-3'	17	dsRNA
S	1mms	2k3f	T. maritima ribosomal protein L11	133	–	23S rRNA fragment	58	dsRNA
S	2err	2cq3	H. sapiens ataxin-2-binding protein 1 (Fox 1)	88	–	5'-UGCAUGU-3'	7	ssRNA
S	2ad9	1sjq	H. sapiens polypyrimidine tract-binding protein RBD1	98	–	5'-CUCUCU-3'	6	ssRNA
S	2adb	1sjr	H. sapiens polypyrimidine tract-binding protein RBD2	127	–	5'-CUCUCU-3'	6	ssRNA
S	2py9	2jzx	H. sapiens poly(rC)-binding protein 2	141	–	Human telomeric RNA	12	ssRNA
S	2adc	2evz	H. sapiens polypyrimidine tract-binding protein RBD3 and 4	208	–	5'-CUCUCU-3'	6	ssRNA
S	3bx2	1m8z	H. sapiens pumilio 1 homology domain	328	–	5'-UGUAUAUUA-3'	9	ssRNA
X	1ekz	1stu	D. melanogaster maternal effect protein	76	–	RNA hairpin	30	dsRNA
X	2hgh	2j7j	S. laevis transcription factor IIIA	87	–	5S rRNA fragment	55	dsRNA
X	1hvu	2vg5	Human immunodeficiency virus HIV1 reverse transcriptase	954	–	RNA pseudoknot	30	dsRNA
X	1b7f	3sxl	D. melanogaster SXL-lethal protein	167	–	5'-GUUGUUUUUUUU-3'	12	ssRNA
X	2c0b	2vmk	E. coli ribonuclease E	488	–	5'-UUUACAGUAUU-3'	11	ssRNA
X	1msw	1aro	E. phage DNA-directed T7 RNA polymerase	863	–	mRNA	47	ssRNA
X	1ob2	1efc	E. coli elongation factor EF-TU	393	1ehz	tRNA-PHE	76	tRNA

TABLE S11 – Codes PDB des 40 complexes protéine-ARN utilisés dans les *benchmarks* I et II, par difficulté (D, avec R pour corps rigide, S pour semi-flexible et X pour flexible), avec une description de la protéine et de l'ARN, ainsi que le nombre d'acides aminés et d'acides nucléiques dans la structure 3D et le code PDB de la structure 3D de la protéine (pdb aa) et de l'ARN (pdb na) non liés si disponibles. Le type d'ARN est aussi indiqué entre simple brin (ssRNA), double brin (dsRNA) et ARN de transfert (tRNA).

	ES				TOP10					TOP100				ROC-AUC			
pdb	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG	Attendu	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG
1asy	0.49	3.69	0.88	0.87	7	8	7	7	7.195	28	95	92	92	0.43	0.74	0.58	0.57
1av6	3.15	1.53	0.00	0.00	5	10	10	10	8.815	77	100	97	97	0.38	0.88	0.62	0.62
1b23	1.47	2.87	0.94	0.94	4	10	9	9	5.037	4	99	88	88	0.39	0.81	0.61	0.61
1c0a	0.46	5.89	2.82	2.76	1	10	9	9	2.568	11	99	75	75	0.30	0.91	0.70	0.70
1ddl	0.52	2.64	1.24	1.24	2	3	1	1	2.789	18	35	28	28	0.48	0.64	0.53	0.52
1dfu	2.11	5.07	0.22	0.22	4	10	10	10	9.24	47	100	99	99	0.28	0.93	0.74	0.73
1di2	0.59	4.05	0.92	0.92	10	10	8	8	7.553	99	98	90	89	0.41	0.91	0.60	0.60
1e8o	1.18	2.08	1.48	1.48	6	2	8	8	3.376	23	37	62	62	0.44	0.63	0.56	0.56
1f7u	0.17	4.46	2.58	2.55	4	10	10	10	4.105	4	94	75	75	0.21	0.89	0.80	0.80
1feu	1.16	1.72	0.11	0.11	4	10	9	9	9.081	65	99	96	96	0.34	0.81	0.67	0.67
1ffy	0.02	7.09	1.11	1.10	0	10	4	4	3.121	0	100	17	17	0.42	0.93	0.58	0.58
1fxl	0.08	6.26	0.26	0.24	0	10	10	10	5.104	0	100	98	98	0.18	0.79	0.83	0.82
1gtf	4.28	1.61	0.00	0.00	10	10	10	10	9.628	97	100	100	100	0.49	0.72	0.51	0.51
1h3e	0.41	5.02	0.96	0.95	0	10	4	4	5.918	11	100	65	65	0.35	0.77	0.65	0.65
1h4s	0.81	1.90	1.72	1.75	1	4	4	4	1.19	2	34	35	35	0.42	0.63	0.58	0.58
1hq1	1.43	2.42	0.68	0.67	6	3	0	0	2.624	31	40	6	6	0.57	0.67	0.43	0.43
1j1u	1.56	1.40	0.15	0.15	10	9	10	10	8.331	96	92	97	97	0.53	0.67	0.47	0.47
1j2b	0.02	6.25	4.17	4.14	0	10	9	9	1.317	0	100	75	75	0.13	0.87	0.87	0.87
1jbs	0.58	0.79	0.59	0.59	0	0	3	3	2.528	1	5	25	26	0.44	0.61	0.56	0.56
1jid	2.52	0.05	0.00	0.00	2	10	10	10	9.357	63	98	95	95	0.37	0.65	0.64	0.63
1k8w	0.59	7.73	2.78	2.70	1	10	10	10	3.916	5	100	86	86	0.35	0.94	0.67	0.65
1knz	0.00	5.68	0.19	0.19	0	10	3	3	3.11	0	100	62	63	0.15	0.93	0.86	0.85
1lng	2.74	3.91	0.77	0.77	6	10	5	5	7.299	70	95	58	58	0.61	0.65	0.39	0.39
1m8v	0.85	1.70	0.51	0.50	10	3	10	10	7.017	90	51	81	81	0.44	0.54	0.56	0.56
1m8x	1.68	4.61	0.26	0.26	7	10	9	9	8.413	86	100	90	90	0.43	0.75	0.58	0.58
1mzp	0.76	3.15	1.96	1.95	0	6	4	4	1.592	11	51	39	38	0.40	0.71	0.60	0.60
1n35	0.10	7.27	2.62	2.62	0	8	0	0	0.732	0	85	13	13	0.34	0.99	0.67	0.67
1n78	0.33	3.67	0.53	0.53	0	10	4	4	4.39	0	98	62	62	0.30	0.92	0.71	0.71
1ooa	3.45	2.56	0.00	0.00	6	10	10	10	8.662	75	100	95	95	0.44	0.91	0.57	0.57
1pgl	3.13	0.00	0.00	0.00	10	9	9	9	9.344	80	98	99	99	0.24	0.85	0.76	0.76
1q2r	0.00	6.33	1.13	1.11	0	10	9	9	4.585	0	100	93	93	0.24	0.82	0.77	0.77
1qf6	0.09	7.30	2.92	2.92	0	10	7	7	1.154	0	99	52	52	0.23	0.90	0.77	0.77

Résultats des fonctions de score atomiqes de la fonction par défaut de RosettaDock (ROS), positive (POS), négative (NEG) et sans contrainte (ALL) pour les complexes de la PRIDB : score d'enrichissement, nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100 et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

	ES				TOP10					TOP100				ROC-AUC			
pdb	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG	Attendu	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG
1qtq	0.05	6.36	2.57	2.54	0	10	9	9	2.971	4	100	80	80	0.30	0.89	0.70	0.70
1r3e	0.66	6.89	2.65	2.65	0	10	6	6	3.356	10	100	75	75	0.41	0.90	0.60	0.60
1r9f	1.55	3.19	0.28	0.28	10	10	10	10	9.359	99	100	98	98	0.38	0.87	0.63	0.62
1sds	0.80	1.51	0.47	0.47	5	6	2	2	4.183	41	37	32	32	0.55	0.59	0.45	0.45
1ser	0.10	4.35	0.51	0.51	0	9	7	7	2.999	0	95	41	39	0.29	0.96	0.72	0.71
1si3	0.27	4.66	0.11	0.09	0	10	10	10	7.681	0	100	100	100	0.08	0.75	0.92	0.92
1t0k	1.53	1.34	0.73	0.73	8	0	4	4	3.725	59	14	27	27	0.55	0.51	0.45	0.45
1tfw	0.02	4.38	3.33	3.31	0	10	1	1	0.225	0	97	14	14	0.22	1.00	0.79	0.78
1u0b	0.02	6.33	3.44	3.41	0	10	7	7	2.616	0	97	69	69	0.21	0.87	0.79	0.79
1un6	0.66	2.93	1.29	1.30	0	9	7	7	3.952	1	93	87	86	0.30	0.83	0.71	0.71
1uvj	0.51	3.20	0.66	0.66	0	6	2	2	2.066	0	73	47	46	0.31	0.80	0.69	0.69
1vfg	2.96	0.32	0.11	0.11	9	10	10	10	9.313	87	99	98	98	0.42	0.74	0.59	0.59
1wpu	2.16	3.48	0.14	0.13	10	10	9	9	8.746	78	100	95	95	0.51	0.62	0.49	0.49
1wsu	0.50	1.72	1.50	1.48	0	2	6	6	4.816	14	32	72	72	0.36	0.52	0.64	0.64
1wz2	0.25	5.18	1.81	1.72	0	10	5	4	2.451	0	98	55	53	0.31	0.79	0.70	0.69
1yvp	2.26	2.69	0.00	0.00	8	10	10	10	9.384	84	100	99	99	0.37	0.84	0.63	0.63
1zbh	0.15	1.74	1.33	1.33	2	8	7	7	3.809	4	69	62	62	0.34	0.68	0.66	0.66
2a8v	0.14	2.20	1.81	1.83	2	8	6	6	4.775	14	84	64	65	0.34	0.53	0.66	0.66
2anr	0.55	2.06	2.14	2.15	7	0	3	3	2.927	23	18	48	48	0.38	0.57	0.62	0.62
2asb	1.18	5.26	0.13	0.13	2	10	6	6	5.475	3	100	60	60	0.46	0.92	0.55	0.55
2az0	0.22	4.33	1.36	1.34	0	10	1	1	1.351	1	77	12	12	0.40	0.84	0.61	0.61
2azx	0.13	2.29	1.85	1.81	0	9	9	9	7.22	10	90	90	89	0.29	0.63	0.72	0.71
2b3j	1.56	3.18	0.11	0.11	0	10	10	10	7.172	7	100	100	100	0.24	0.93	0.76	0.76
2bgg	0.30	4.89	2.25	2.21	0	10	10	10	5.42	4	100	98	97	0.28	0.88	0.73	0.73
2bh2	0.29	7.08	1.75	1.74	0	10	8	8	2.34	0	100	82	82	0.32	0.89	0.68	0.68
2bte	0.98	2.15	0.23	0.23	7	10	6	6	7.251	76	98	83	82	0.42	0.82	0.59	0.58
2bu1	0.18	2.41	1.75	1.73	5	3	2	2	3.909	17	54	43	43	0.38	0.60	0.62	0.62
2bx2	0.07	2.08	0.98	0.96	0	5	3	2	3.441	0	42	21	20	0.36	0.81	0.64	0.64
2ct8	0.68	3.09	0.87	0.87	3	10	4	4	5.817	37	87	69	69	0.35	0.79	0.65	0.65
2czj	3.12	3.51	0.06	0.05	10	10	10	10	9.135	94	100	97	97	0.51	0.78	0.49	0.49
2d6f	2.30	2.75	0.00	0.00	10	10	1	1	8.458	81	96	64	64	0.63	0.67	0.37	0.37
2der	0.88	4.52	0.88	0.87	0	10	5	5	3.535	0	100	84	84	0.26	0.96	0.75	0.74

Résultats des fonctions de score atomiqes de la fonction par défaut de RosettaDock (ROS), positive (POS), négative (NEG) et sans contrainte (ALL) pour les complexes de la PRIDB : score d'enrichissement, nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100 et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

	ES				TOP10					TOP100				ROC-AUC			
pdb	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG	Attendu	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG
2du3	2.14	2.11	0.05	0.05	9	6	10	10	9.16	65	84	95	95	0.48	0.52	0.52	0.52
2e9t	0.00	7.70	1.83	1.80	0	10	4	4	1.688	0	100	75	73	0.19	1.00	0.83	0.82
2f8k	1.37	1.47	0.92	0.92	9	1	4	4	4.993	78	32	63	62	0.51	0.57	0.49	0.49
2f8s	1.27	2.19	0.62	0.62	6	10	10	10	6.802	53	100	93	93	0.40	0.67	0.60	0.60
2fk6	0.84	2.93	0.61	0.61	10	10	9	9	8.17	78	98	94	94	0.46	0.86	0.54	0.54
2fmt	0.59	5.99	1.07	1.06	0	10	7	7	2.675	10	100	44	43	0.40	0.90	0.61	0.61
2gic	1.11	3.06	0.13	0.13	0	8	3	3	3.956	0	87	36	35	0.38	0.92	0.63	0.63
2gje	1.11	2.60	0.27	0.26	6	9	10	10	7.233	74	95	93	93	0.30	0.83	0.70	0.70
2gjw	0.11	4.25	0.18	0.18	8	10	10	10	7.806	28	100	100	100	0.22	0.89	0.79	0.79
2gtt	1.03	3.38	0.41	0.38	0	9	0	0	2.419	0	88	12	12	0.43	0.88	0.57	0.57
2gxb	0.54	1.87	0.98	0.95	0	3	3	3	4.715	16	45	37	37	0.49	0.59	0.51	0.51
2hw8	1.89	3.72	0.14	0.14	9	10	9	9	7.302	23	99	91	90	0.52	0.93	0.49	0.48
2i82	3.45	7.14	0.05	0.05	6	10	10	10	6.908	60	100	96	96	0.41	0.84	0.60	0.60
2iy5	0.07	6.69	2.12	2.10	0	4	0	0	0.92	0	79	3	3	0.31	0.96	0.70	0.70
2jlv	0.53	0.74	0.00	0.00	0	1	0	0	4.519	0	13	7	7	0.33	0.81	0.67	0.67
2nqp	0.74	3.02	0.77	0.77	0	5	1	1	2.642	5	58	25	25	0.43	0.74	0.57	0.57
2nug	0.29	6.56	5.83	5.82	0	10	0	0	0.365	0	100	1	1	0.66	1.00	0.40	0.37
2ozb	1.10	6.82	2.24	2.23	1	10	10	10	3.716	3	100	95	95	0.37	0.84	0.63	0.63
2pjp	0.01	1.75	2.16	2.17	1	1	6	6	2.983	1	33	53	53	0.31	0.60	0.69	0.69
2po1	1.98	1.35	0.00	0.00	10	10	1	1	9.139	97	100	55	55	0.45	0.75	0.56	0.56
2qux	0.16	3.18	1.63	1.61	1	4	0	0	1.94	1	56	16	17	0.40	0.70	0.60	0.60
2r7r	0.03	4.54	0.16	0.16	0	10	6	6	4.231	0	98	74	73	0.18	0.91	0.83	0.83
2r8s	1.35	1.96	0.28	0.28	8	10	9	9	7.263	42	90	83	82	0.46	0.59	0.54	0.54
2vnu	0.09	2.20	1.13	1.13	0	2	0	0	2.481	0	58	30	30	0.24	0.93	0.76	0.76
2voo	0.08	1.69	0.13	0.13	10	10	10	10	7.556	96	100	97	97	0.43	0.85	0.58	0.57
2w2h	0.50	3.96	2.71	2.70	2	10	10	10	5.472	20	99	97	97	0.36	0.81	0.64	0.64
2wj8	0.06	6.11	0.66	0.65	0	10	5	5	2.429	0	99	52	52	0.22	0.91	0.79	0.79
2z2q	0.49	2.64	1.36	1.36	3	9	5	5	5.908	26	89	76	76	0.41	0.60	0.59	0.59
2zi0	3.01	0.48	0.01	0.01	0	9	4	4	7.917	10	88	58	58	0.27	0.80	0.74	0.74
2zko	1.15	4.04	0.99	1.00	0	7	0	0	1.338	9	83	9	9	0.57	0.83	0.44	0.43
2zni	0.30	6.44	1.70	1.64	0	10	9	9	2.671	0	100	48	46	0.40	0.76	0.61	0.61
2zue	0.30	4.53	1.50	1.49	0	10	9	9	4.929	0	99	82	82	0.22	0.89	0.79	0.78

Résultats des fonctions de score atomiqes de la fonction par défaut de RosettaDock (ROS), positive (POS), négative (NEG) et sans contrainte (ALL) pour les complexes de la PRIDB : score d'enrichissement, nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100 et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.

	ES				TOP10					TOP100				ROC-AUC			
pdb	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG	Attendu	ROS	POS	ALL	NEG	ROS	POS	ALL	NEG
2zzm	0.11	8.06	3.62	3.57	0	10	6	6	1.256	0	100	67	67	0.18	0.93	0.83	0.83
3a6p	6.30	7.17	0.00	0.00	0	7	0	0	1.972	54	96	0	0	0.80	0.99	0.21	0.20
3bso	0.94	5.84	0.08	0.08	0	10	0	0	2.472	1	100	0	0	0.43	0.99	0.58	0.58
3bt7	0.57	4.98	0.75	0.72	1	10	10	10	5.376	20	100	90	90	0.33	0.79	0.68	0.68
3ciy	1.71	2.79	0.19	0.18	4	10	2	2	5.629	44	82	54	53	0.59	0.69	0.41	0.41
3d2s	0.04	0.79	0.56	0.56	6	10	9	9	8.929	62	97	97	97	0.32	0.71	0.69	0.69
3dd2	0.74	2.00	1.21	1.21	0	3	2	2	2.468	4	43	34	35	0.44	0.66	0.56	0.56
3egz	1.05	3.05	0.41	0.39	6	10	7	7	5.223	6	97	91	90	0.16	0.79	0.84	0.84
3eph	0.19	8.81	5.65	5.61	0	10	10	10	1.002	0	100	90	90	0.13	0.99	0.88	0.87
3eqt	0.40	7.19	0.50	0.48	2	10	8	8	5.188	54	100	85	85	0.38	0.94	0.62	0.62
3ex7	0.05	4.06	0.55	0.53	0	10	10	10	7.516	0	96	99	99	0.18	0.75	0.83	0.82
3fht	2.19	3.66	0.01	0.01	0	10	10	10	8.25	0	100	100	100	0.18	0.84	0.83	0.83
3foz	0.48	8.00	2.67	2.66	0	10	8	8	1.341	0	100	48	48	0.21	0.98	0.80	0.79
3gib	0.37	1.80	1.78	1.78	0	6	6	6	2.324	0	48	56	57	0.31	0.49	0.69	0.69
3hax	0.04	6.51	4.73	4.71	1	10	9	9	1.027	1	100	81	80	0.19	0.88	0.81	0.81
3hl2	2.67	3.66	0.00	0.00	0	10	10	10	9.062	56	100	100	100	0.48	0.99	0.54	0.53
3htx	0.02	8.04	3.70	3.69	0	10	6	6	1.045	0	100	52	51	0.22	0.96	0.79	0.78
3i5x	0.31	3.55	1.20	1.19	0	10	8	8	6.791	4	100	80	79	0.35	0.80	0.65	0.65
3iab	0.59	5.28	1.17	1.13	0	10	10	10	4.77	7	92	100	100	0.24	0.73	0.77	0.77
3icq	0.14	2.83	1.12	1.12	0	10	7	8	5.14	10	88	80	80	0.35	0.81	0.65	0.65
3iev	1.30	5.64	0.94	0.92	8	10	10	10	7.914	67	100	99	99	0.61	0.65	0.40	0.40
3k62	2.51	3.40	0.15	0.15	9	10	9	9	8.482	86	100	96	96	0.35	0.78	0.65	0.65
3l25	0.35	7.02	1.09	1.07	0	7	2	2	1.205	0	79	35	33	0.38	0.99	0.64	0.62
3snp	0.02	6.63	2.80	2.77	1	10	10	10	3.201	1	100	86	86	0.25	0.84	0.75	0.75

TABLE S12 – Résultats des fonctions de score atomiqes de la fonction par défaut de RosettaDock (ROS), positive (POS), négative (NEG) et sans contrainte (ALL) pour les complexes de la PRIDB : score d'enrichissement, nombre de presque-natifs dans le top10, nombre de presque-natifs attendus en moyenne par un tri aléatoire, nombre de presque-natifs dans le top100 et aire sous la courbe ROC (ROC-AUC) sur 10 000 candidats par pdb évalué.