École Doctorale INTERFACES
Approches interdisciplinaires: fondements, applications et innovations

Titre de la thèse

THÈSE

présentée et soutenue publiquement le XXX

pour l'obtention du

Doctorat de l'Université Paris-Saclay

spécialité: Les sciences du vivant

par

M. David Mignon

Composition du jury

Rapporteurs :	Dr. Prénom1 Nom1	Rapporteur externe
	Dr. Prénom2 Nom2	Rapporteur externe
	Pr. Prénom3 Nom3	Rapporteur interne
Examinateurs :	Dr. Prénom4 Nom4	Examinateur
	Dr. Prénom5 Nom5	Directeur de thèse
	Dr. Prénom6 Nom6	Directeur de thèse

Laboratoire de XXX

Remerciements

XXX

à XXX.

Table des matières

Liste des figures

Liste des tables

Abreviations

H algorithme heuristique

MC algorithme Monte-Carlo

RE algorithme “Replica Exchange”’

GMEC “’Global minimal energie cost”

Pfam ”Protein family databank”

Introduction

XXX

Contexte

XXX
XXX
Citation entre crochets [**??**].
Citation dans le texte **?**.

Chapitre 1

CPD

Chapitre 2

proteus

Chapitre 3

Méthodes

3.1 Les méthodes pratiques

Nous cherchons maintenant à déterminer les performances et les qualités des différents algorithmes de proteus. Pour évaluer les différents algorithmes de proteus, comme pour leur établir un paramétrage, nous effectuons des séries de tests. Grâce à l'algorithme de type toulbar2 il est possible d'obtenir la séquence/conformation qui possède la plus haute énergie de dépliement. Cela constitue une information important qui va nous servir d'élément de comparaison.Le facteur temps est également un élément déterminant. Il est dans certain cas limitant, nous ne savons pas à l'avance quand toulbar2 termine. Et il apparaît d'emblée illusoire d'espérer voir ce programme converger dans toutes les situations intéressantes dans un temps raisonnable.D'autres métriques qui caractérisent les séquences d'acides aminés de meilleurs énergies obtenues seront également utilisées pour les évaluations et pour les paramétrages.

Dans la suite, on appelle «position active», une position pour laquelle, tous les types d'acides et tous les rotamères de chaque type d'acide aminé sont autorisés, au court de la recherche de proteus. On désigne «séquence/conformation» une séquence d'acides aminés munie à chaque position d'un rotamère (le backbone étant de toute façon fixé).Tandis ce que le terme simple «séquence» sans plus de précision désigne une séquence d'acides aminés.

3.1.1 les protéines

Les tests sont effectués sur neuf protéines choisies pour avoir des longueurs de backbone variées, plusieurs domaines représentés, mais aussi plusieurs structures pour chaque famille présente. Ainsi l'ensemble se décompose en deux protéines SH3 de 56 et 57 résidus, de trois protéines PDZ de longueur comprise entre 82 et 97 résidus et enfin de trois protéines

SH2 longues de 105 ou 109 résidus.L'ensemble a une moyenne, arrondie à l'unité inférieure, de quatre-vingt-neuf positions, voir les détails en table 3.1.

Code PDB	résidus	nombre de positions	domaine
1ABO	64-119	56	SH3
1CKA	134-190	57	SH3
1R6J	192-273	82	PDZ
1G9O	9-99	91	PDZ
2BYG	186-282	97	PDZ
1BM2	55-152	98	SH2
1O4C	1-105	105	SH2
1M61	4-112	109	SH2
1A81	9-117	109	SH2

Table 3.1 – Les protéines

Protein	1ABO	1CKA
1ABO	100 (6e-42)	26 (1e-07)
1CKA	26 (1e-07)	100 (2e-41)

Table 3.2 – Pourcentage d'identité et e-value des alignements Blast native vs native pour nos protéine SH3.

Protein	1R6J	1G9O	2BYG
1R6J	100(1e-59)	25 (3e-07)	no
1G9O	25 (3e-07)	100(2e-66)	35 (2e-11)
2BYG	no	35 (2e-11)	100(7e-71)

Table 3.3 – Pourcentage d'identité et e-value des alignements Blast native vs native pour nos protéine PDZ (no= pas de touche avec une e-value inférieure à 10).

Alignements Blast croisés

Protein	1BM2	1O4C	1M61	1A81
1BM2	100 (7e-74)	36 (2e-16)	38(6e-10)	35 (1e-13)
1O4C	36 (2e-16)	100(2e-79)	27(3e-10)	33 (2e-12)
1M61	38 (6e-10)	27 (3e-10)	100(6e-81)	57 (2e-47)
1A81	35 (1e-13)	33 (2e-12)	57(2e-47)	100(5e-83)

Table 3.4 – Pourcentage d'identité et e-value des alignements Blast native vs native pour nos protéine SH2.

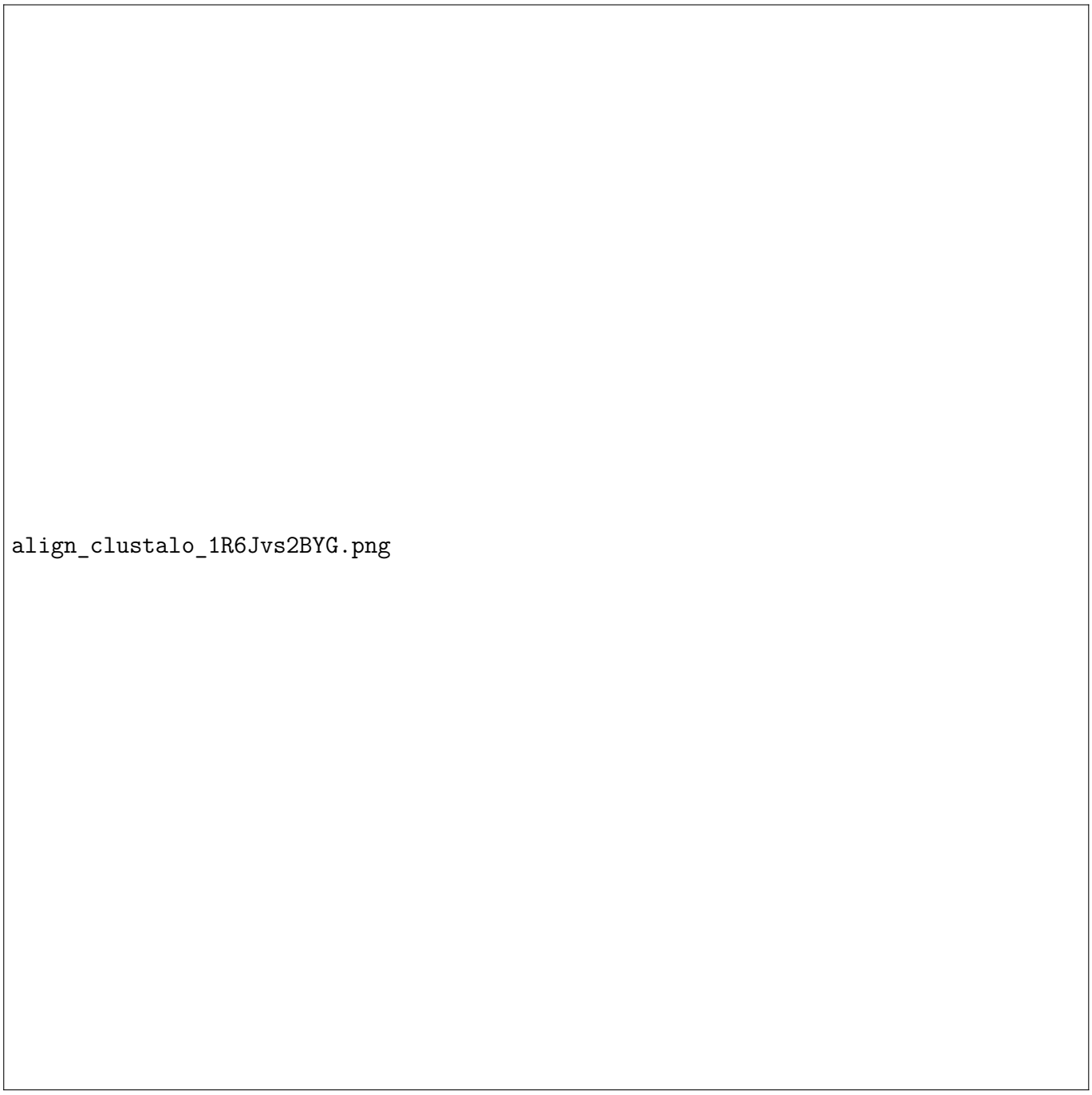

Figure 3.1 – alignement de 1R6J et 2BYG obtenu avec Clustal Omega version 1.2.1

3.1.2 Description des tests

Les tests sont répartis en deux ensembles :

1. un ensemble de tests où toutes les positions de la séquence sont actives (cela correspond aux situations de design complet de protéines)
2. un ensemble de tests où le nombre de positions actives est gardé sous contrôle de façon à maîtriser la taille de l'espace d'exploration

Ensemble «Tout actif» Pour le premier ensemble de tests,la totalité de la matrice d'énergie est exploitée et pour chaque position l'espace d'exploration correspond à l'espace d'état déclaré dans le fichier ".bb".C'est-à-dire que tous les types de résidu et tous les rotamères sont possibles à chaque position. Comme l'espace des séquences/conformations à explorer est gigantesque, nous ne faisons pas de tentatives de recherche du GMEC par méthode exacte.

Nous effectuons des recherches avec les algorithmes suivants :

— heuristique, noté H par la suite ;
— Monte-Carlo, noté MC ;
— «Replica Exchange», noté RE) ;

L'ensemble «nombre d'actifs limité» L'ensemble «Nombre d'actifs limité» est composé de six groupes de tests avec un nombre de positions actives fixe définit de la façon suivante :

1. aucune position active
2. une position active
3. cinq positions
4. dix positions
5. vingt positions
6. trente positions

Lorsqu'une position n'est pas active,l'acide aminé de la position est fixé en utilisant l'acide aminé de la séquence native. La chaîne latérale est, elle, laissée libre. Il n'y a donc jamais dans nos tests de position où l'état est complètement fixé.

Le groupe «aucune position active» n'est constitué que d'un test par algorithme pour chaque protéine. Il y a donc neuf tests par algorithme. Ce sont les tests pendant lesquels la séquence d'acides aminés est fixe et correspond à la séquence native de la protéine.

Pour les tests avec une seule position active, comme des temps de calcul le permettent, nous décidons d'être exhaustifs : Toutes les positions sont testées, il y a alors huit cent quatre tests par algorithme. Pour tous les autres groupes de tests (cinq,dix,vingt et trente positions actives), cinq tests sont effectués par protéine, c'est-à-dire quarante-cinq tests par algorithme.

le choix des positions actives Pour définir complètement les tests, il reste maintenant à décrire le choix des positions actives pour les groupes de numéro trois jusqu'à six. Il y a peu d'intérêt à tester des situations avec des positions actives sans interaction entre-elles. En effet, s'il existe une position active P dont chaque résidu est sans interaction avec tous les résidus possibles des autres positions actives, déterminer le meilleur état pour P est proche du test du groupe 2 avec P comme position active. Toutefois, cela n'est pas exactement la même question, parce que les positions actives différentes de P peuvent influencer la position de la chaîne latérale de positions inactives qui à leur tour peuvent influencer l'état de P. Ainsi, le choix des positions actives ne se fait non pas par tirage aléatoire, car le risque d'obtenir des positions avec peu d'interactions est trop grand. Il se fait sous contrainte d'interaction.

positions en interactions Pour cela, nous utilisons la notion de voisinage de proteus. Elle se définit de la façon suivante : Deux positions P et Q sont en interactions s'il existe un rotamère r_P de P et un rotamère r_Q de Q tels que :

$$|E(r_P,r_Q)| > S_{Vois}$$

avec S_{Vois} un seuil donné par l'utilisateur à la configuration de proteus (voir chap. ? ? pour les détails).

Alors on appelle «n-uplet en interaction» la donnée de n positions avec $n \in \{5,10,20,30\}$ et d'un seuil S_{Vois} tels que pour toute paire de positions (P,Q) du n-uplet, P et Q sont en interactions.

choix des positions actives Pour définir les positions actives,nous exécutons proteus en mode verbeux, sans effectuer d'optimisation.Pour cela, il existe plusieurs façons de procéder, ici nous utilisons le mode Monte-Carlo avec une trajectoire de zéro pas. Ces exécutions produisent en sortie standard la liste des voisins pour chaque position au seuil donnée en paramètre. Pour chacune des neuf protéines, nous exécutons proteus avec S_{Vois} égal à dix, cinq et un à tour de rôle ; trois listes de voisins sont obtenues. Ensuite,un script dédié recherche dans ces listes, les n-uplets en interaction, en partant de la liste de voisins

au sens le plus fort, c'est-à-dire dix, vers celle au sens le plus faible (0.1).La recherche s'arrête lorsque cinq n-uplets au moins sont trouvés.

Nous obtenons quarante-cinq n-uplets pour le groupe à cinq (respectivement dix, vingt et trente) positions actives pour un seuil S_{Vois} égal à dix (respectivement dix, un et un). Les positions actives de tous les tests sont en annexe **??**).Pour chaque n-uplet, un fichier de configuration de proteus est créé dans lequel la balise <Space_Constraints> fixe les positions inactives en utilisant le type d'acide aminé présent dans la séquence native.

3.1.3 Définition de protocole comparable

Nous voulons comparer les algorithmes très différents. Un algorithme peut garantir l'obtention du minimum global en énergie (GMEC) si l'exécution se termine, mais ne garantit pas qu'elle se termine. Un autre permet un contrôle très fin du temps d'exécution sans garantie du GMEC, et d'autres enfin ont des objectifs plus large que la seule obtention du GMEC. Mais le GMEC reste le meilleur point de commun. Nous allons donc y concentrer une part importante des comparaisons.

Nous devons noter également que l'obtention du GMEC est théorique, en pratique nous n'avons pas de preuve que le code de l'algorithme exact que nous utilisons n'a pas de bogue. Cependant,nous mettons de côté cette éventualité et dans toute la suite GMEC désigne aussi bien le minimum global en énergie que le résultat de toulbar2 lorsqu'il se termine. Le Monte-Carlo et le «Replica exchange» possèdent de nombreux paramètres de configuration, ce qui rend l'ensemble des protocoles possibles très grand. Se pose alors la question de l'optimisation du protocole. L'objectif fixé ici,n'est pas la recherche d'un protocole optimal pour chacun des tests, mais d'évaluer, avec les tests, un protocole optimisé par algorithme. Nous allons alors dans un premier temps, recherche les meilleurs paramétrages pour le Monte-Carlo et le «Replica Exchange» sur l'ensemble de tests «tout actif». Puis, sur la base des résultats obtenus, les protocoles seront fixés pour effectuer les comparaisons sur l'ensemble «tout actif» et celui à «nombre d'actifs limité». Le programme toulbar2 possède aussi de nombreuses options. Deux paramétrages différents seront utilisés.

Pour rendre les protocoles comparables, le temps d'exécution maximum est fixé à vingt-quatre heures pour tous les exécutions. Toulbar2 donne sa meilleure séquence/conformation en dernier, il n'y a donc pas post-traitement nécessaire. C'est également le cas pour le Monte-Carlo à condition de configurer l'impression de la trajectoire avec la balise $Print\_Threshold = 0.$ dans le fichier de configuration. Pour le "Replica Exchange" et l'heuristique, un tri des séquences selon l'énergie est nécessaire. Mais il n'y a pas beaucoup de séquences :

1. L' Heuristique fournit une séquence/conformation à chaque cycle.
2. Le "Replica Exchange avec $Print\_Threshold = 0$ produit autant de fichiers de séquences/rotamères que de marcheurs.Chacun ne contenant pas plus de quelques dizaines de séquences/rotamères.

Nous pouvons donc négliger la durée du tri dans le temps total d'exécution.

Protocole heuristique Pour l'algorithme heuristique, il n'y a dans notre situation qu'un seul paramètre à renseigner : le nombre de cycles à effectuer. Quelques essais préliminaires sur la plus grosse protéine (Table 3.1) avec toute les positions actives, montre que la version utilisée de proteus peut effectuer jusqu'à environ 110000 cycles sur nos machines de calculs en l'espace de vingt-quatre heures. Ainsi, le protocole H est défini comme le protocole qui utilise le mode heuristique de proteus et qui effectue cent dix mille cycles. Sont également définis les variantes H-, H+ et H++ comme des protocoles plus courts ou plus longs à facteur entier près (Table 3.5). Par ailleurs, certaines comparaisons de l'heuristique avec le Monte-Carlo ont été faites avec une version précédente du programme proteus.Ce protocole sera noté h. Il diffère aussi de H par le fait que l'option d'optimisation du compilateur Intel utilisé est -O2 contre -O3 pour H.

Nom	nombre de cycles
H	110000
H-	1100
H+	330000
H++	990000
h	100000

Table 3.5 – Les protocoles heuristiques

Protocoles Monte-Carlo On distingue deux ensembles de protocoles Monte-Carlo.Dans le premier, les noms sont de la forme "mc*". Il rassemble les protocoles utilisés pour le paramétrage du Monte-Carlo.Le second est constitué des protocoles utilisés lors des comparaisons.

Les éléments à paramétrer pour l'algorithme Monte-Carlo sont les suivants :

1. la température
2. le nombre de pas (avec le nombre de trajectoires et la longueur de trajectoire)
3. Le seuil de voisinage
4. Les probabilités de changements de la séquence/conformation

Ce qui représente un ensemble de protocoles trop grand pour une approche exhaustive. Pour l'essentiel, nous allons faire varier les paramètres un par un, en prenant comme point de départ un protocole qui rend le comportement de marcheur Monte-Carlo «proche» de l'heuristique.

La température est le paramètre principal du Monte-Carlo, c'est elle qui contrôle le taux d'acceptation du critère de Metropolis.Alors,la première étape de cette optimisation va consister à faire varier la température , entre 0.001 et 0.5 , en conservant les autres paramètres fixés (protocoles de mc0 à mc5).Le nombre de pas total effectué est le produit de deux paramètres, le nombre de trajectoires et la longueur de trajectoire. Les protocoles mc1b et mc2b testent l'effet d'une augmentation du nombre de pas. Tandis que mc2c et mc2d testent l'effet de la variation du nombre de trajectoires par rapport à la longueur. Le protocole mc2e s'intéresse aux probabilités de changement de la trajectoire. Il y a cinq balises dans proteus qui contrôle ces changements :

<Prot> donne la probabilité de modifications de rotamère à une position.

<Prot_Prot> donne la probabilité de modifications de rotamère à deux positions.

<Mut> donne la probabilité de modifications de type de résidu à une position.

<Mut_Prot> donne la probabilité de modifications de rotamères à deux positions.

<Mut_Mut> donne la probabilité de modifications de type de résidu à deux positions.

La table 3.6 donne les probabilités utilisées par ces cinq paramètres dans l'ordre de la liste précédente.

Enfin, mc4b se distingue des autres par un seuil de voisinage plus grand ((Table 3.6)).

Seconde version de proteus Pour la partie comparaison avec les autres algorithmes, quatre protocoles sont utilisés. Les protocoles MCa et MCb s'inspirent fortement de mc2d et mc2e , en étant adapté à la contrainte du temps de calcul de la comparaison et en utilisant la nouvelle version de proteus (les lettres capitales dans le nom des protocoles signifient l'utilisation de la dernière version de proteus). MCa- est une variante de MCa avec une trajectoire six fois plus courte.Enfin, MC0 s'inspire de mc0 dans le sens où la température est suffisamment froide pour que nous puissions considérer qu'il n'y a pas de baisse de l'énergie au cours d'une trajectoire.

Protocoles "Replica Exchange" L'algorithme «Replica Exchange» (RE) est une extension du Monte-Carlo. Les paramètres d'un protocole RE sont ceux d'un protocole Monte-Carlo plus trois autres :

Nom	Temp	Long. de trajectoire(mega)	Nb de trajectoires	Voisin	Proba
mc0	0.001	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc1	0.1	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc2	0.2	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc3	0.3	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc4	0.5	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc5	0.7	3	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc1b	0.1	6	1000	10	1 ; 1 ; 1 ; 1 ;0
mc2b	0.2	6	1000	10	0 ; 1 ; 0.1 ; 0 ;0
mc2c	0.2	3	10000	10	0 ; 1 ; 0.1 ; 0 ;0
mc2d	0.2	3000	1	10	0 ; 1 ; 0.1 ; 0 ;0
mc2e	0.2	3	1000	10	1 ; 0 ; 0.1 ; 0 ;0
mc4b	0.5	10	100	10	0 ; 1 ; 0 ; 1 ;0
MC0	0.01	1000	1	10	1 ; 0 ; 0.1 ; 0 ;0
MCa	0.2	6000	1	10	1 ; 0 ; 0.1 ; 0 ;0
MCa-	0.2	1000	1	10	1 ; 0 ; 0.1 ; 0 ;0
MCb	0.2	6000	1	10	0 ; 1 ; 0.1 ; 0 ;0

Table 3.6 – Les protocoles Monte-Carlo

— le nombre de marcheurs
— la température pour chaque marcheur
— la période de «swap», c'est-à-dire la période (en nombre de pas) à laquelle le test de Hasting sur l'échange de température est effectué.

Pour avoir des exécutions en parallèle avec au plus un marcheur par coeur du processeur, nous limiter nos tests à quatre ou huit marcheurs. La distribution des températures est un élément déterminant dans le comportement des marcheurs, car c'est elle qui pilote en grande partie le taux d'acceptation des échanges de températures. Nous suivons l'idée proposée par Kofke de lui faire suivre une progression géométrique ($\frac{T_i}{T_{i+1}} = C$, avec C une constante) [**???**]. Ceci garantie alors que le taux d'acceptation d'échange entre $T_i et T_{i+1}$ soit égale pour tout nos i.De plus, nous souhaitons centrer approximativement, nos distributions sur la température ambiante (environ 0.6 kcal/mol). Dans toute la suite, les températures et les énergies sont exprimées en kcal/mol.

Voici les températures pour le RE quatre marcheurs :

— 10, 1, 0.1 et 0.01
— 2, 1, 0.5 et 0.25
— 1, 0.5, 0.25 et 0.125

,et celles pour le RE huit marcheurs :

— 3 , 2 , 1.333 , 0.888 , 0.592 , 0.395 , 0.263 et 0.175

— 10 , 3.16 , 1 , 0.316 , 0.1 , 0.0316 , 0.01 et 0.00316

Ici les protocoles ne se font qu'avec une seule trajectoire par marcheur. Et la contrainte du temps de calcul se comprend comme vingt-quatre heures de calculs cumulées sur tous les marcheurs. Ainsi les longueurs de trajectoire sont définit pour le RE à quatre marcheurs comme le quart d'une trajectoire MC, pour le RE à huit marcheurs comme le huitième.

La table 3.7 donne les probabilités utilisées par les cinq balises qui contrôlent les modifications de la séquence/conformation à chaque pas, dans l'ordre de la liste de la section 3.1.3.

Nom	marcheurs	Temp	Traj (mega)	seuil voisin	Proba	swap period (me
RE4a	4	10<->0.01	1500	10	1 ; 0 ; 0.1 ; 0 ;0	7.5
RE4b	4	1<->0.125	1500	10	1 ; 0 ; 0.1 ; 0 ;0	7.5
RE4c	4	2<->0.25	1500	10	1 ; 0 ; 0.1 ; 0 ;0	7.5
RE8a1	8	10<->0.00316	750	0	1 ; 0 ; 0.1 ; 0 ;0	2.5
RE8a2	8	10<->0.00316	750	10	1 ; 0 ; 0.1 ; 0 ;0	2.5
RE8b1	8	3<->0.175	750	10	0 ; 1 ; 0.1 ; 0 ;0	7.5
RE8b2	8	3<->0.175	750	10	1 ; 0 ; 0.1 ; 0 ;0	7.5
RE8b3	8	3<->0.175	750	10	1 ; 0 ; 0.1 ; 0 ;0	1

Table 3.7 – Les protocoles «Replica Exchange»

Protocoles Toulbar2 Après avoir converti nos matrices au format «wcsp» grâce à un script dédié,nous pouvons utiliser toulbar2. Le protocole de recherche du GMEC est le suivant : L'exécutable toulbar2 de version 0.9.7.0 est lancé avec les options « -l=3 -m -d : -s», ce qui correspond au paramétrage conseillé dans la documentation CDP [**??**]. Si l'exécution se termine en moins de vingt-quatre heures, le protocole est achevé. Sinon le programme est arrêté et une seconde version (la 0.9.6.0) est lancée avec les options «-l=1 -dee=1 -m -d : -s». Au bout de vingt-quatre heures si le programme n'est pas terminé, il est arrêté. La dernière séquence/conformation imprimée en sortie est collectée. Le choix de la seconde version et du paramétrage fait suite à une discussion avec monsieur Seydou Traoré.

Toulbar2 offre également la possibilité de fournir la liste des séquences/conformations dont l'énergie est comprise entre celle qui correspond au GMEC, E_{GMEC} et une autre E_{upper_bound} donnée en paramètre. Pour utiliser cette fonctionnalité nous utilisons le paramétrage : «-d : -a -s -ub=E_{upper_bound} ».Cependant, il s'avère que cette utilisation peut utiliser une quantité de mémoire vive importante.Alors, pour eviter tout plantage de nos machines, la mémoire que toulbar2 peut allouer est limité à 30 Go.

3.1.4 Outils d'analyse des données

Superfamily/SCOP Superfamily [?] est un ensemble composé :

- D'une base de données de modèles de Markov cachés, où chaque modèle représente une structure 3D d'un domaine de la classification SCOP.
- D'une série de scripts qui annotent à partir des informations de la base,les séquences données en entrée. Ici, nous utilisons uniquement l'association au modèle 3D le plus vraisemblable.

Nous travaillons avec la base de données à la version 1.75, et en conjonction, nous utilisons SAM (version 3.5) [?] et HMMER (version 3.0) [?] recommandés par l'équipe de Superfamily. Le paramétrage utilisé est celui par défaut.

Taux d'identité de séquences Soient S et N deux séquences d'acides aminés de même longueur l.

Le Taux d'identité $Id(S,N)$ de S par rapport N est égal au pourcentage de position où l'acide aminé est identique dans S et N. C'est-à-dire

$Id(S,N) = \frac{\sum_{1<i<l} \mathbb{1}(s_i,n_i)}{l} \times 100$

avec s_i et n_i l'acide animé en i de S et de N respectivement, et $\mathbb{1}(x,y)$ la fonction qui vaut 1 lorsque x=y et 0 sinon.

Taux d'identité par position Le taux d'identité d'un alignement A_S à la position i par rapport à une séquence N de même longueur se définit comme :

$Id(A_S,i) = \frac{\sum_{1<j<m} \mathbb{1}(s_i^j,n_i)}{m} \times 100$, avec m le nombre de séquences de A_S.

Alignements Pfam Ce taux d'identité donne une mesure de la ressemblance entre un alignement et une séquence. Cela nous permet de comparer nos séquences calculées à la séquence native. Mais cela n'est pas notre seule objectif.Et nous voulons les évaluer par rapport à l'ensemble des séquences du domaine protéique de la native. La base de données Pfam (Protein families database) [?] regroupe les domaines protéiques connus en famille. Chaque famille étant représentée par des alignements multiples de séquences et des profiles de modèles de Markov cachés [?]. Dans la suite, nous n'utiliserons l'alignement dit « seed» qui se base sur un petit ensemble de membres représentatifs de la famille et l'alignement « full» , plus large, qui est généré par modèle de Markov caché à partir de l'alignement «seed». Les alignements correspondent pour nous aux familles PF00017 (domaine SH2), PF00018 (domaine SH3) et PF00595 (domaine PDZ).

Score BLOSUM Pour tenir compte des ressemblances et des différences entre les acides aminés lors d'une substitution, nous avons besoin d'une matrice de coût.Nous utilisons la matrice BLOSUM62 (BLOcks SUbstitution Matrix) [**?**] qui est construite à partir de blocs d'alignement très conservés (ici plus de 62% d'identités).Les fréquences des mutations y sont calculées.Le score BLOSUM d'une substitution est alors le logarithme de la fréquence de la mutation correspondante.À cela est ajouté un score de pénalités pour l'insertion d'un gap (c'est-à-dire un saut dans l'alignement).

On définit alors simplement un score de similarité de deux séquences de même longueur comme la somme des scores BLOSUM62 sur toutes les positions. De même le score de similarité d'un alignement par rapport à une séquence sera défini comme la moyenne des scores de similarité sur ensemble des séquences de l'alignement. Et enfin un score de similarité de deux ensembles de séquences alignés entre eux comme la moyenne des scores de similarité du premier ensemble par rapport aux séquences du second.

similarité d'un ensemble à une famille Pfam Afin de calculer un score de similarité d'un ensemble de nos séquences par rapport à une famille Pfam, il faut commencer par aligner nos séquences avec l'alignement de la famille.Pour cela nous utilisons le programme d'alignement BLAST [**?**].Il implémente une heuristique qui recherche puis étend les meilleurs alignements locaux. Nous procédons comme suit :

1. La commande blastpgp est utilisée avec comme database (paramètre -d) l'alignement Pfam et comme séquence en entrée (paramètre -i) la séquence native.
2. Dans la sortie blast, la séquence qui produit l'alignement le plus significatif avec la native est collectée, notons-la S_0.
3. L'alignement blast est alors utilisé pour positionner la native par rapport à S_0 et les gaps nécessaires pour aligner la native à S_0 sont ajoutés.
4. Le positionnement et les gaps sont alors appliqués tels quels à la liste de nos séquences.

Répartition de l'énergie selon les centiles Pour étudier différentes distributions d'ensemble de séquences/conformations selon l'énergie,nous déterminons les centiles de la façon suivante :

1. L'ensemble de séquences/conformations est trié selon l'énergie.
2. L'intervalle entre la meilleure énergie et la moins bonne est divisé en cent intervalles consécutifs contenant le même nombre de séquences/conformations (un centième du cardinal de l'ensemble).

3. Les quatre-vingt-dix-neuf valeurs d'énergie obtenues par ce découpage sont les centiles.

Chapitre 4

Comparing stochastic search algorithms for computational protein design

David Mignon and Thomas Simonson*

†Laboratoire de Biochimie (UMR CNRS 7654), Dept. of Biology, Ecole Polytechnique, Palaiseau, France *Corresponding author. Email : thomas.simonson@polytechnique.fr

Abstract

Computational protein design depends critically on an energy function, a conformation space model, and an algorithm to search the space of sequences and conformations. We compare three search algorithms that are stochastic : a heuristic method, a Monte Carlo method (MC), and a Replica Exchange Monte Carlo method (REMC) that propagates several walkers, or replicas at different temperatures. The methods are applied to nine test proteins, with 1, 5, 10, 20, 30, or all amino acid positions allowed to mutate, for a total of about 200 tests. Results are also compared to a recent, exact, "Cost Function Network" method (CFN) that identifies the global minimum energy conformation (GMEC) in favorable cases. The designed sequences accurately reproduce the experimental amino acid types for positions in each protein's hydrophobic core. The heuristic and REMC methods are in good mutual agreement, yielding very similar optimal solutions ; they accurately reproduce the GMEC when it is known, with a few exceptions. Plain MC performs well for most but not all cases, occasionally departing from the GMEC by 3–4 kcal/mol. With REMC, the diversity of the sequences sampled, as measured by the mean sequence entropy, agrees well with exact enumeration in the range where the latter is possible : about 2 kcal/mol above the GMEC. Beyond this range, room temperature

walkers routinely sample sequences up to 10 kcal/mol above the GMEC, providing thermal averages and at least a rough solution to the inverse protein folding problem.

4.1 Introduction

Computational protein design (CPD) has developed into an important tool for biotechnology **??????**. Starting from a 3D structural model, CPD explores a large space of possible sequences and conformations, to identify protein variants that have certain predefined properties, such as stability or ligand binding. Conformational space is usually defined by a library of sidechain rotamers, which can be discrete or continuous, and by a finite set of backbone conformations or a specific repertoire of allowed backbone deformations. The energy function usually combines physical and empirical terms **???**. Both solvent and the unfolded protein state are described implicitly.

The number of amino acid positions that are allowed to mutate can vary, depending on the problem of interest, from 2 or 3 to several dozen. Thus, the combinatorial complexity can be enormous, so that speed is important, as well as accuracy. In addition, it is usually important to identify not one but several high-scoring sequences, for at least three reasons. First, if the typical error in the energy function is σ_E, we should enumerate all the possible sequences/structures within one or two σ_E of the optimal one. Second, it may be of interest to characterize the diversity of a sequence family, by enumerating sets of sequences compatible with its backbone fold (the "inverse folding problem") **?????**. Third, we may want to compute properties that are averaged over structural and possibly sequence fluctuations at a given temperature, which requires that we explore solutions within the thermal range. An example is the calculation of ligand binding constants, following a method introduced recently **?**. Calculation of acid/base constants by constant-pH Monte Carlo is another example, which can also be seen as a subproblem of CPD, where sidechain protonation state changes are treated as mutations **???**.

Thus, the complexity and cost of a CPD calculation will depend on several factors. While energy calculations usually represent the bulk of the cost, the power and efficiency of the exploration method are also very important. Several exploration methods exist that can identify exactly the global minimum energy sequence and conformation, or GMEC. These include "dead end elimination" methods, or DEE **??**, branch-and-bound methods **??**, and cost function network methods **??**. While some of these methods can handle large problems, they usually cannot enumerate suboptimal solutions within a large interval σ_E above the GMEC (more than a few kcal/mol). Partly for this reason, stochastic methods remain popular, such as Monte Carlo (MC) **??**. MC has two advantages : with an appropriate setup, it samples sequences/conformations from a known, Boltzmann distribution, and it can be readily combined with enhanced sampling methods developed in the broader field of molecular simulations, such as Replica Exchange or umbrella sampling **??**.

Our goal here is to assess three stochasttic exploration methods for a series of CPD problems of increasing complexity. The first method is a heuristic method that is not guaranteed to find the global minimum energy conformation, or GMEC, but has been effective in applications **???**. The second method is a Monte Carlo (MC) exploration, which samples sequences/conformations from a Boltzmann distribution **???**. The third is an enhanced, multi-walker MC, which performs "replica exchange" **???**. Several walkers, or replicas are propagated at different temperatures, and exchange conformations at regular intervals according to a MC test. We refer to it as REMC. These methods are also compared to a fourth method that is exact, in the sense that it can provably identify the GMEC in favorable cases **??**. It is based on 'cost function networks", or CFN, where the cost function is the energy, and the network refers to the set of interacting amino acids. The CFN method uses a depth-first branch-and-bound search through a tree of rotamer assignments, with fast integer arithmetic for the energy evaluations. It can also enumerate all the sequence/conformation combinations within a given energy range δE (not too large) above the GMEC. It is implemented in the Toulbar2 program, by Schiex and coworkers. Other exact methods exist, some of which appear to be even faster than CFN. Our goal, however, is not to "rank" the stochastic and exact methods, but rather to compare our three stochastic methods to each other, and this is facilitated if an exact enumeration of low energy states has also been done.

We use a CPD model that is rather simple but representative of a large class of applications. We use a discrete set of sidechain rotamers, a fixed backbone structure, and we assume that the energy function is pairwise additive ; i.e., the energy has the form of a sum over residue pairs **???**. With these simplifications, all possible residue pair interactions can be computed ahead of time and stored in a lookup table **?** ; exploration is then done in a second stage. Thus, the cost of energy calculations and sequence/structure exploration are well-separated. The method is implemented in the Proteus CPD package **??** (except for the CFN sequence exploration, done with Toulbar2). Our MC framework is presented in some detail below ; the other methods are recalled more briefly. < We considered nine test proteins from three structural classes : SH3, SH2, and PDZ domains. For each one, we chose different numbers and sets of residues to mutate and we applied the different exploration methods, using several possible parameterizations for each one. To characterize the different methods, we compared their speed, their ability to identify the GMEC, and their sampling of suboptimal sequences/conformations above the GMEC. The designed sequences were characterized by computing their similarity to natural sequences, their classification by the Superfamily fold recognition tool **??**, and their sequence entropies. For the few cases where there were large differences between the methods (several kcal/mol

between best-scoring sequences), the 3D structural models were compared. Overall, the heuristic method is the most successful in identifying low energy solutions, while REMC is almost as successful but has the advantage of sampling from a Boltzmann distribution over a large energy range, yielding thermal averages.

4.2 Methods

4.2.1 Monte Carlo : general framework

We consider a polypeptide of n amino acids. Its sequence S is written S $= t_1 t_2 \cdots t_n$, where t_i is the sidechain type of amino acid i. We assume that each amino acid i can take on a few different types t, t', ... that form a set T_i. For each sequence, there are two classes of structures : folded and unfolded. For the folded form, all the sequences S share the same, precise geometry for the polypeptide backbone ; only the sidechain positions can vary. Specifically, the sidechain of each amino acid i can explore a few discrete conformations or "rotamers" r, r', ... (around 10 per type t_i). The structure of the unfolded form is not specified ; the energy is assumed to be independent of the particular unfolded structure, and to have the additive form :

$$E_u(S) = \sum_{i=1}^{n} E_u(t_i) = \sum_{i=1}^{n} \left(e_u(t_i) - kT \log n_u(t_i) \right), \tag{4.1}$$

where $E_u(t_i)$ is a free energy associated with sidechain type t_i in the unfolded state, and the rightmost form separates it into an energy component $e_u(t_i)$ and a conformational entropy term, where kT is the thermal energy and $n_u(t_i)$ is the number of conformations or rotamers available to sidechain type t_i in the unfolded state.

We perform a Monte Carlo simulation **???** where one copy of the folded protein is explicitly represented. The unfolded state is included implicitly, by propagating the simulation with the energy function $E_M = E_f - E_u$ (the folding energy). One possible elementary MC move is to change a rotamer r_i in the current folded sequence ; the energy change is $\Delta E_M = \Delta E_f = E(...t_i,r'_i...) - E(...t_i,r_i...)$. Another possible move is a mutation : we modify the sidechain type $t_i \to t'_i$ at a chosen position i in the folded protein, assigning a particular rotamer r'_i to the new sidechain. The energy change is

$$\Delta E_M = \Delta E_f - \Delta E_u = (E_f(...t'_i,r'_i...) - E_f(...t_i,r_i...)) - (E_u(t'_i) - E_u(t_i)) \tag{4.2}$$

ΔE_M measures the stability change due to the mutation (for the given set of rotamers) ; it is as if we performed the reverse mutation $t'_i \to t_i$ in the unfolded form.

If the moves are generated and accepted with an appropriate Metropolis-like scheme, the Markov chain will visit states according to their Boltzmann probability :

$$p_M(S,c) = \frac{e^{-\beta(E_f(S,c)-E_u(S))}}{\sum_{S'}\sum_{c'} e^{-\beta(E_f(S',c')-E_u(S'))}} \tag{4.3}$$

where $\beta = 1/kT$ and the subscript M indicates probabilities sampled by the Markov chain. For two conformations c, c' of sequence S, the Markov probability ratio is $p_M(S,c)/p_M(S,c') = e^{-\beta(E_f(S,c)-E_f(S,c'))}$. For two sequences S, S', the probability ratio is

$$\frac{p_M(S)}{p_M(S')} = \frac{\sum_c e^{-\beta(E_f(S,c)-E_u(S))}}{\sum_{c'} e^{-\beta(E_f(S',c')-E_u(S'))}} = \frac{e^{-\beta\Delta G_{\text{fold}}(S)}}{e^{-\beta\Delta G_{\text{fold}}(S')}} \tag{4.4}$$

In the ratio of Markov probabilities, we recognize the ratio of Boltzmann factors for S and S' folding, so that we have the second equality, where $\Delta G_{\text{fold}}(S)$ denotes the folding free energy of sequence S (respectively, S').

Eq. (4.4) has a simple interpretation : the Markov chain, with the chosen energy function $E_M = E_f - E_u$ and appropriate move probabilities, leads to the same distribution of states as a macroscopic, equilibrium, physical system where all sequences S, S', ... are present at equal concentrations, and are distributed between their folded and unfolded states according to their relative stabilities. This is exactly the experimental system we want our Markov chain to mimic. In this interpretation, a Monte Carlo mutation move S $\rightarrow$ S' amounts to unfolding one copy of S and refolding one copy of S'.

It remains to specify the move generation probabilities and choose an appropriate acceptance scheme **???**. Let $\alpha(o \rightarrow n)$ be the probability to select a trial move between two states o and n and $acc(o \rightarrow n)$ the probability to accept it. If the simulation obeys detailed balance, we have

$$N(o)\pi(o \rightarrow n) = N(n)\pi(n \rightarrow o), \tag{4.5}$$

where $N(o)$, $N(n)$ are the equilibrium populations of states o and n. With "ergodic" move sets such as the one used here (see below), detailed balance is guaranteed in the limit of a very long simulation. To produce Boltzmann statistics, we choose the acceptance probabilities **???** :

$$acc(o \rightarrow n) = \exp(-\beta\Delta E_M)\frac{\alpha(n \rightarrow o)}{\alpha(o \rightarrow n)} \quad \text{if } \Delta E_M > 0; 1 \text{ otherwise} \tag{4.6}$$

where $\Delta E_M = E_M(n) - E_M(o)$ is the o $\rightarrow$ n energy difference.

For a rotamer move at a particular position in the polypeptide chain, of type t, we define the move generation probability as $\alpha(o \to n) = \frac{1}{n_f(t)} = \alpha(o \to n)$; all possible choices for the new rotamer are equiprobable, forward and backward rotamer moves have the same generation probability, and Eq. (4.6) reduces to the simple Metropolis test **?**.

For a mutation move at a particular position, we define $\alpha(o \to n)$ as follows :
(a) select a new type t' with equal probabilities $\alpha_t(o \to n) = \frac{1}{N}$ for all N possible types ;
(b) choose a rotamer r' for the new sidechain with equal probabilities $\alpha_r(o \to n) = \frac{1}{n_f(t')}$ for all $n_f(t')$ possible folded state rotamers.
The overall probability is therefore

$$\alpha(o \to n) = \alpha_t(o \to n)\alpha_r(o \to n) = \frac{1}{N n_f(t')} \tag{4.7}$$

The $o \to n$ and $n \to o$ probabilities are different whenever the old and new sidechain types have different numbers of possible rotamers. With these move probabilities, the mutation acceptance probability can be written :

$$\begin{aligned} acc(t \to t') &= e^{-\beta(\Delta E_f - \Delta E_u)} \frac{n_f(t)}{n_f(t')} = e^{-\beta(\Delta E_f - \Delta e_u)} \frac{n_f(t) n_u(t')}{n_u(t) n_f(t')} \quad \text{if } \Delta E_M > 0 \quad (4.8) \\ &= 1 \quad \text{otherwise} \quad (4.9) \end{aligned}$$

If the number of rotamers in the folded and unfolded states are the same, $n_u = n_f$, the fraction on the right will cancel out. However, the rotamer numbers also appear in the energy change that determines whether the move is uphill, ΔE_M.

With REMC, several simulations ("replicas" or "walkers") are propagated in parallel, at different temperatures ; periodic swaps are attempted between two walkers's conformations. The swap is accepted with probability

$$acc(t \to t') = \text{Min}\left[1, e^{(\beta_i - \beta_j)(\Delta E_i - \Delta E_j)}\right] \tag{4.10}$$

where β_i, β_j are the inverse temperatures of the two walkers and ΔE_i, ΔE_j are their folding energies **??**.

4.2.2 MC and REMC : implementation details

For plain MC, we use one- and two-position moves, where either rotamers, types, or both are changed. For two-position moves, the second position is selected among those that have a significant interaction energy with the first one, 10 kcal/mol or more. For REMC, we use four or eight walkers, with thermal energies kT_i that range from 0.125 to 2 or 3

kcal/mol, and are spaced in a geometric progression : T_{i+1}/T_i = constant, following Kofke **?**. Conformation swaps are attempted at regular intervals, between walkers at adjacent temperatures. The precise parameter settings are given in Table 4.1.

4.2.3 Heuristic sequence optimization

The heuristic sequence optimization uses an iterative minimization **??**. One "heuristic cycle" proceeds as follows : an initial amino acid sequence and set of sidechain rotamers are chosen randomly. These are improved in a stepwise way. At a given amino acid position i, the best amino acid type and rotamer are selected, with the rest of the sequence held fixed. The same is done for the following position $i + 1$, and so on, performing multiple passes over the amino acid sequence until the energy no longer improves or a set, large number of passes is reached (500 passes). The final sequence, rotamer set, and energy are output, ending the cycle. The method can be viewed as a steepest descent minimization, starting from a random sequence, and leading to a nearby, local, (folding) energy minimum. Below, we typically perform ∼100.000 heuristic cycles for each protein, thus sampling a large number of local minima on the energy surface.

4.2.4 Cost function network method

The CFN method is implemented in the Toulbar2 program **??**. The Proteus energy matrices are converted to the Toulbar format with a perl script. With this format, all the interaction energies are approximated as positive integers, without loss of generality. We used Toulbar2 version 0.9.7.0 with a recommended parameterization (options -l=3 -m -d : -s) ; for the unsuccessful cases (GMEC not identified) we systematically repeated calculations with version 0.9.6.0 and a more aggressive protocol (options -l=1 -dee=1 -m -d : -s). To enumerate sequence/conformation pairs that have energies higher than the GMEC, Toulbar2 is run with the "suboptimal" option and an energy threshold. Available memory was limited to 30 gigabytes.

4.2.5 Energy function

The energy function has the form :

$$E = E_{\text{bonds}} + E_{\text{angles}} + E_{\text{dihe}} + E_{\text{impr}} + E_{\text{vdw}} + E_{\text{Coul}} + E_{\text{solv}} \tag{4.11}$$

The first six terms represent the protein internal energy. They were taken from the Charmm19 empirical energy function **?**. The last term represents the contribution of

solvent. We used a "Coulomb + Accessible Surface Area", or CASA implicit solvent model **??** :

$$E_{\text{solv}} = \left(\frac{1}{\epsilon} - 1\right) E_{\text{Coul}} + \sum_i \sigma_i A_i \tag{4.12}$$

Here, ϵ is a dielectric constant that scales the Coulomb energy set 23, following earlier tests **?**. A_i is the solvent accessible surface area of atom i ; σ_i is a parameter that reflects each atom's preference to be exposed or hidden from solvent. The solute atoms were divided into 4 groups with the following σ_i values (cal/mol/Å^2) : unpolar (-5), aromatic (-40), polar (-8) and ionic (-10). Hydrogen atoms were assigned a surface coefficient of zero. Surface areas were computed by the Lee and Richards algorithm **?**, using a 1.5 Å probe radius. Pairwise additivity errors for the surface energy term were corrected by applying a reduction factor of 0.5 to buried pairs **??**. Energy calculations were done with a modified version of the Xplor program **??**.

The energies $E_u(t)$ associated with the unfolded state were determined empirically to give reasonable amino acid compositions for the protein families considered here **?** ; they are reported in Supplementary Material.

4.2.6 Test systems and preparation

We considered nine protein domains, from the SH3, SH2, and PDZ families, listed in Table 4.2. Each domain is known to fold stably and has an associated crystal structure used for our calculations. Systems were prepared and energy matrices computed using procedures described previously **??**. Briefly, each PDB structure was minimized through 200 steps of conjugate gradient minimization. For each residue pair, interaction energies were computed after 15 steps of energy minimization, with the backbone fixed and only the interactions of the pair with each other and the backbone included. Sidechain rotamers were described by a slightly expanded version of the library of Tuffery et al **?**, which has a total of 228 rotamers (sum over all amino acid types).

4.2.7 Sequence characterization

Designed sequences were compared to the Pfam alignment for the corresponding family, using the Blosum40 scoring matrix and a gap penalty of -6. Each Pfam sequence was also compared to its own Pfam alignment. For these Pfam/Pfam comparisons, if a test protein T was part of the Pfam alignment, the T/T self comparison was left out, to be more consistent with the designed/Pfam comparisons. If the test protein T was not part of the Pfam alignment, we used Blast to identify its closest Pfam homologue H

and left the T/H comparison out, for consistency. The Pfam alignments were either the "seed" alignment for each family (around 50 sequences) or much larger, "full" alignments, with 6287, 3052, and 14944 sequences, respectively, for the SH3, SH2, and PDZ families. Similarities were computed for protein core residues, defined by their near-complete burial, and listed in Results.

Designed sequences were submitted to the Superfamily library of Hidden Markov Models **??**, which attempts to classify sequences according to the SCOP classification **?**. Classification was based on SCOP version 1.75 and version 3.5 of the Superfamily tools. Superfamily executes the hmmscan program, which implements a Hidden Markov model for each SCOP family and superfamily ; here hmmscan was executed with an E-value threshold of 10^{-10}, using a total of 15438 models to represent the SCOP database.

To compare the diversity in the designed sequences with the diversity in natural sequences, we used a standard, position-dependent sequence entropy **?**, computed as follows :

$$S_i = -\sum_{i=1}^{6} f_j(i) ln f_j(i) \tag{4.13}$$

where $f_j(i)$ is the frequency of residue type j at position i, either in the designed sequences or in the natural sequences (organized into a multiple alignment). Instead of the usual, 20 amino acid types, we employ six residue types, corresponding to the following groups : {LVIMC}, {FYW}, {G}, {ASTP}, {EDNQ}, an {KRH}. This classification was obtained by a cluster analysis of the BLOSUM62 matrix **?**, and also by analyzing residue-residue contact energies in proteins **?**. To get a sense of how many amino acid types appear at a specific position i, we usually report the residue entropy in its exponentiated form, $\exp(S_i)$, which ranges from 1 to 6.

4.3 Results

Our main focus is to characterize sequence/structure exploration methods and their ability to sample low energy sequences. We begin, however, by showing that the sequences we sample are similar to experimental ones. Indeed, the performance of exploration methods depends on the shape and ruggedness of the energy surface, and should be tested in situations where the energy function is sufficiently realistic, as judged by the quality of the designed sequences. After that, we compare the ability of the four exploration methods to identify low energy sequences, including the GMEC. Finally, we consider the diversity of sequence sets, or density of states sampled by each method.

4.3.1 Quality of the designed sequences

We first report information on the quality of our designed sequences. We use sets of REMC sequences to illustrate the main features. The best REMC protocol, REMCD (Table 4.1) was used. Results with the other exploration methods are expected to be similar. Indeed, while the methods sometimes exhibit differences of up to a few kcal/mol between their best sequences, the average sequence quality of the 100-1000 best sequences is typically similar between methods. Table 4.3 summarizes results for our 9 test proteins in design calculations where all positions were allowed to change types. All mutations were allowed, except mutations to/from Gly and Pro, since these are likely to change the backbone structure. REMC was done with 8 replicas at temperatures between 0.175 and 3 kT units. Simulation lengths were 750 million steps (per replica). The top 10000 sequence/conformation combinations were retained, corresponding to 200–400 unique sequences. For the 1A81 SH3 domain, none of these sequences was recognized by Superfamily as an SH3 family, or even superfamily member. For the other 8 proteins, all the retained sequences were recognized as members of the correct superfamily *and* family, with match lengths and E-values given in Table 4.3. Thus, our designed sequences are largely similar to experimental ones. Sequence identities to wildtype (including 1A81) are 31% on average (Table 4.3), similar to earlier studies with the same energy function **??**. Representative sequence logos for the protein core are shown in Fig. 1, illustrating the agreement between designed and experimental sequences. Similarity scores were computed between the designed sequences and experimental sequences from the Pfam database **?**. For the protein core region, the similarity is similar to that between experimental sequences, as shown in Fig. 2.

4.3.2 Finding the GMEC

CPU and memory limits for each method

The ability of an exploration method to sample low energy sequences depends on the CPU and memory ressources available, as well as on detailed parameterization choices. Here, we set somewhat arbitrary limits, to remain within a practical run situation. For CFN, we set a maximum time limit of 24 hours and a memory limit of 30 gbytes. For the heuristic method, we used 110,000 heuristic cycles, increased to 330,000 or 990,000 cycles in a few cases ; even for these cases, run times did not exceed 24 hours. For MC, we ran up to 10^9 simulation steps, which corresponded to CPU times of 9 hours at most. For REMC, we ran 0.75 10^9 simulation steps per replica, with a few exceptions. We used an OpenMP, shared memory parallelization on a single processor, with one replica per core.

Total CPU time per core was never more than 3 hours, for a total CPU use of less than 24 hours. For the heuristic, MC, and REMC methods, memory requirements are modest ; about 2 gbytes for the largest calculations. Run times are shown in Fig. 3 as a function of the number of designed positions, which varies from one position to the entire protein (about 90 positions). For comparison, the CPU time needed to compute the energy matrix for a single pair of designed positions, using an advanced energy function (Generalized Born solvent plus a sophisticated surface area term) and a single core of a recent Intel processor is about 5 hours.

The MC and REMC methods require choices of move probabilities and temperatures, which affect the sampling in ways that vary from protein to protein. Fig. 4 shows the lowest energy sampled with a small collection of protocols : one heuristic, one MC, and five REMC protocols, applied to our nine proteins, with all positions allowed to mutate (except Gly/Pro). The protocol details are given in Table 4.1. For these large design problems, the GMEC is not known. Instead, for each protein, the overall best energy (the best of the seven protocols) is taken as the reference, or zero value. For a given protein, the best energy varies by up to 12 kcal/mol from one protocol to another (compare the 1BM2 REMCa and REMCc energies or the 1CKA MC and REMCd energies). For each protocol, the best energy obtained was averaged over the nine proteins, giving a mean "error" ; these values are also reported in Fig. 4. The lowest mean error is 1.0 kcal/mol, with REMCd. In other words, REMCd gives a best energy that is, on average, 1.0 kcal/mol above the overall best energy. Based on these and other similar tests, for the rest of this work, we used the specific MC protocol in Table 4.1 and the REMCd replica exchange protocol, which are generally good but not necessarily optimal for every situation.

Optimal sequences/structures with up to 10 designed positions

As our first series of tests, we did calculations for each test protein with zero, one, or five designed positions. Results are summarized in Fig. 5. With zero positions, only rotamers are optimized (at all positions in the protein). With one, we systematically designed each position of each protein in turn (plus rotamers at all positions). With 5, we picked the positions randomly, close together in the structure, in 5 different ways, for a total of 45 tests. Two positions were considered close by if they have at least one rotamer combination that gives an interaction energy of 10 kcal/mol or more (in absolute magnitude ; eg, steric overlap). In all these cases, CFN found the GMEC very rapidly (seconds) ; the heuristic also found the GMEC, with much longer run times (an hour). MC found the GMEC in all but a few cases (Fig. 5), with run times of a few minutes.

As a second series of tests, we chose randomly for each protein a set of ten positions to design ; the other positions had fixed types but explored all possible rotamers. The selected positions were close by in the protein structure. For each protein, we made five separate choices of positions to design, for a total of 45 test cases. The CFN, heuristic, and MC methods were run for all 45 cases ; REMC was run only when MC gave a poor result (6 cases, involving 5 proteins). Results are summarized in Fig. 5 and Table 4.4. 20 cases where all methods found the GMEC are not listed in the Table, leaving 25 where at least one method did not find the GMEC. CFN performed very well : only in one case did it not find the GMEC. The lowest energy was sampled in this case with the heuristic, and the best CFN energy was 5.7 kcal/mol higher (despite using the more aggressive CFN protocol).

The heuristic performed about as well as CFN for 10-position design. In one case, CFN did not find the GMEC and the heuristic gave the lowest energy (2BYG-1). In 39 cases, the heuristic found the GMEC. In 3 cases, it was within 0.15 kcal/mol of the GMEC, with no mutations (only rotamer differences). In one case (1CKA-5), it was 0.29 kcal/mol above the GMEC, with no mutations. Tripling the number of heuristic cycles allowed the GMEC to be reached (within 0.07 kcal/mol) in all these cases, with run times below 6 hours. There was only one real failure, 1M61-2, where the best heuristic solution was 3.5 kcal/mol above the GMEC, with 3 mutations relative to the GMEC. For this case, the GMEC was recovered (within 0.01 kcal/mol) if the number of cycles was increased to 990,000, for a run time of 7 hours. Switching from the heuristic structure (after 330,000 cycles) to the GMEC requires concerted changes in 3 adjacent sidechain positions. This is only possible during a heuristic cycle if there is a downhill, connecting pathway made of single position changes, which is evidently very rare for this particular test. Thus, the heuristic method can only find the GMEC if it draws the right combination of types/rotamers at the very beginning of a cycle ; hence the need for 990,000 cycles.

Plain MC did slightly less well for 10-position design. In 21 cases, it found the GMEC. In 18 cases, its best sequence was within 0.2 kcal/mol of the GMEC, with 0–3 mutations (one on average). Notice that 0.2 kcal/mol is the thermal energy for the MC protocol employed. In 6 cases, its best sequence was between 0.9 and 4.5 kcal/mol above the GMEC, with 2–7 mutations (3 on average). For these 6 cases, REMC was run, and sampled sequences within 0.40 kcal/mol of the GMEC, except for one case where its best sequence was 0.80 kcal/mol above the GMEC. Overall, MC or REMC reached the GMEC to within 0.40 kcal/mol in all but one case. A 0.40 kcal/mol energy difference is actually less than the average pairwise additivity errors in the energy function **???**, and so one might consider this performance to be about as good as the CFN and heuristic methods. In terms of

speed, for 10-position design, all the methods were comparable (a few hours per run on average).

Optimal sequences with 20 or 30 designed positions

We did similar tests with 20 designed positions, selected randomly in 5 different ways for each protein, as above. Results are given in Fig. 5 and Table 4.5. CFN found the GMEC in 28 out of 45 cases ; in 2 others, it found the best energy of the 4 methods. For 6 of these 30 cases, the more aggressive protocol was necessary, and run times were 2–22 hours (11 on average). For 14 of the other 15 cases, the best CFN energy was 0.1–7.5 kcal/mol above the best solution found by the other methods, and 2.8 kcal/mol on average, despite using the more aggressive protocol. For the worst case, the CFN energy was 13.9 kcal/mol above the best solution.

The heuristic method found the GMEC in 22 of the 28 cases where it is known. For the other 6 cases, it was within 0.40 kcal/mol of the GMEC, with 0–4 mutations (2.7 on average). For the 17 cases where the GMEC was not identified by CFN, the heuristic produced the lowest energy of all methods, except one case (104C-1) where it was 0.35 kcal/mol above CFN. Overall, the heuristic either found the best energy of the four methods or was within 0.40 kcal/mol of the best energy.

MC converged to the best energy in 11 cases ; in 25 other cases, it was within 0.50 kcal/mol of the best energy. In the other 9 cases, its best energy was at most 3.2 kcal/mol above the best energy (sampled by the heuristic and/or CFN). Finally, REMC was done for all the test cases. In 6 cases, its best energy was more than 0.50 kcal/mol from the best energy. However, the differences were notably smaller than for plain MC, with an average of just 0.8 kcal/mol for the 6 worst cases and a maximum (for 1G90-1) of 1.25 kcal/mol.

The same tests were done with 30 designed positions ; see Fig. 5 and Table 4.5. CFN found the GMEC in just one case ; in 5 others, it did not find the GMEC but gave the lowest energy overall. In 4 other cases, it was within 0.50 kcal/mol of the best energy sampled by the other methods. For the other 35 cases, its best energy was higher than the best method, with differences of 10 kcal/mol or more in 20 cases.

The heuristic produced the lowest energy in all but 4 cases, with differences in those cases of 0.01, 0.10, 0.70, and 1.69 kcal/mol from the best energy. In the last two cases, CFN produced the best energy. Plain MC found the best energy in only 12 cases, but gave only moderate energy errors : in just 4 cases was its best sequence more than 2 kcal/mol above the overall best energy (differences of 2.2, 2.5, 2.8, and 7.7 kcal/mol). REMC was applied to the 13 cases where the MC errors were largest ; in 4 of these it reduced the error to 0.6 kcal/mol or less. The largest MC error was reduced from 7.7 to 2.4 kcal/mol.

Doubling the REMC trajectory length reduced the two largest remaining errors to 1.1 and 1.8 kcal/mol.

4.3.3 Density of states above the GMEC

The exact CFN method can enumerate exhaustively sequence/conformation states above the GMEC, up to a given energy threshold, if the threshold is not too large. Monte Carlo and REMC explore states randomly, within a typical energy range that depends on temperature. To characterize the diversity of the sequence ensembles, we focus on the CFN and REMC methods, and we consider both the sequence entropy and the total number of states.

The mean, exponentiated sequence entropies $\langle e^{S_i} \rangle$ are reported in Table 4.6 for each test protein. The sequence entropies for the corresponding Pfam alignments (both seed and full) are also shown. The values are averaged over the designed positions in the protein chain, and can be interpreted as a mean number of amino acid classes sampled at each position. There are six classes (see Methods), one of which (Gly) is not available to the designed positions but is present in Pfam. The entropies are much smaller in the designed sets than in the Pfam sets. Retaining the top 10,000 designed sequences, CPD samples 1.3 to 1.7 amino acid classes at each position on average, compared to 3–4 in the Pfam alignments. Thus the CPD sets are much less diverse than Pfam, as observed earlier for these and other protein families **??**. However, we showed earlier that if we did CPD for around ten backbone conformations, corresponding to ten representatives of a particular domain class (SH3, SH2, PDZ), then collected the sampled sequences, the overall entropy was similar to Pfam **??**.

The entropy $S(E)$ is shown in Fig. 6 for the 1CKA SH3 protein as a function of the energy threshold E. Exact CFN results are compared to REMC. For this small protein, complete enumeration was feasible up to an energy threshold of $E = 2$ kcal/mol above the GMEC. REMC samples energies up to about 14 kcal/mol above the GMEC. The REMC sampling is essentilly complete up to about 0.75 kcal/mol above the GMEC, at which point the REMC curve (grey) begins to depart from the exact, CFN curve (black). However, the REMC diversity at each position agrees very well with the CFN result up to about 1.5 kcal/mol above the GMEC. At $E = 2$ kcal/mol above the GMEC, the REMC entropy is still 93% of the exact value. Thus, REMC samples the full sequence diversity at each position in this range, even though it does not sample exhaustively all the combinations of mutations (let alone rotamers) at all positions.

As we consider higher energy threshold values, $E \geq 3$ kcal/mol, the number of states sampled by REMC increases exponentially and the entropy increases in a quasilinear way. Different replicas sample different energy ranges, as expected ; for example, the kT=0.592 and kT=0.888 kcal/mol replicas sample the 4–10 and 11–14 kcal/mol ranges, respectively.

4.4 Conclusions

Stochastic search methods are very common in CPD. They can be extended to very large search spaces that include backbone flexibility **???**, and they do not directly depend on additive energy functions (although additivity can dramatically increase efficiency). In contrast, while exact methods like CFN have been extended beyond purely additive energy functions **??** and discrete rotamer sets **??**, they are less transferable than MC and REMC. REMC is also a powerful engine to explore sequence diversity, with energy ranges of 10 kcal/mol or more above the GMEC readily accessible in this work.

Here, we tested three stochastic search methods, and compared their ability to identify low energy sequences in problems of increasing size/complexity. Direct comparison to the exact GMEC was possible routinely for problems involving up to 10 designed positions, and for 28 of 45 tests with 20 designed positions. The 10-position designs involved total rotamer numbers of 2500–3000 ; for the 20- and 30-position designs, there are about 2000 and 4000 more rotamers, respectively. The larger tests are relevant for whole protein design projects, as well as projects that redesign one or more large protein surfaces (eg, protein crystal design). For these tests, the GMEC was usually not available, and so the stochastic methods could only be evaluated indirectly. Here, the indirect quality indicators were consistency between the methods and general sequence quality compared to experiment. Indeed, agreement between the heuristic and REMC solutions was very good in general, and agreement with experimental Pfam sequences was excellent for core residues, as observed previously with the same energy function (but a heuristic exploration method) **??**. Results with a more advanced protein force field and Generalized Born solvent are expected to be even better **??**. Designed surface residues were less similar to experiment, which is at least partly because the experimental sequences are subject to additional constraints and selective pressures (such as domain-domain interactions), not included in the design.

Overall, the heuristic and REMC methods gave very good agreement with each other and with the GMEC when available. CFN, in its Toulbar2 implementation was very effective for up to 10 designed positions. Exploration speed was similar for all methods for 10-position design, and similar for the stochastic methods applied to the larger problems.

With 8-walker REMC and 0.75 billion steps per walker, the three largest departures from the overall best energy, among 67 difficult, large tests, were 4, 3, and 2.4 kcal/mol. Sequence diversity was also recapitulated accurately, compared to exact enumeration. We have recently extended the method to allow backbone moves, with the help of a hybrid move scheme to be described elsewhere. Overall, both the heuristic and REMC method appear to be effective search methods for all problem sizes.

Acknowledgements

We thank Seydou Traoré for help with the Toulbar2 program and Georgios Archontis, Isabelle André, and Sophie Barbe for helpful discussions.

Table 4.1 – Selected MC and REMC protocols

name	walker temperature(s) kT (kcal/mol)	trajectory length (steps)	move probabilities[a] rot ;mut ;mut+rot ; mut+mut ;	walker swap periodicity[b]
MC	0.2	6 10^9	0 ; 1 ; 0.1 ; 0	-
REMCa	0.125 ; 0.25 ; 0.5 ; 1	1.5 10^9	1 ; 0 ; 0.1 ; 0	7.5 10^6
REMCb	0.25 ; 0.5 ; 1 ; 2	1.5 10^9	1 ; 0 ; 0.1 ; 0	7.5 10^6
REMCc	0.175 ; 0.263 ; 0.395 ; 0.592 ; 0.888 ; 1.333 ; 2 ; 3	0.75 10^9	0 ; 1 ; 0.1 ; 0	7.5 10^6
REMCd	0.175 ; 0.263 ; 0.395 ; 0.592 ; 0.888 ; 1.333 ; 2 ; 3	0.75 10^9	1 ; 0 ; 0.1 ; 0	7.5 10^6
REMCe	0.175 ; 0.263 ; 0.395 ; 0.592 ; 0.888 ; 1.333 ; 2 ;	0.75 10^9	1 ; 0 ; 0.1 ; 0	10^6

[a]Probabilities at each MC step to change, respectively, a rotamer ; a sidechain type ; a type at one position and a rotamer at another ; a type at two positions. [b]The interval between attempts to exchange states between two walkers (using a Metropolis test).

Table 4.2 – Test proteins

type	PDB	length	acronym	type	PDB	length	acronym
PDZ	1G9O	91	NHERF	SH2	1A81	108	Syk kinase
PDZ	1R6J	82	syntenin	SH2	1BM2	98	Grb2
PDZ	2BYG	97	DLH2	SH2	1M61	109	Zap70
SH3	1ABO	58	Abl	SH2	1O4C	104	Src kinase
SH3	1CKA	57	c-Crk				

Table 4.3 – Designed sequence quality measures

	Number of	Identity	Superfamily tests				
	sequences	% to	Match	Superfamily	Superfamily	Family	Family
Protein	tested	wildtype	length	E-value	success rate	E-value	success rate
1A81	236	27	none				
1ABO	203	32	51/58	4.4e-4	100%	2.8e-3	100%
1BM2	209	27	78/98	4.2e-5	100%	2.6e-3	100%
1CKA	416	33	40/57	1.1e-5	100%	3.4e-3	100%
1G9O	338	36	79/91	7.0e-7	100%	2.5e-3	100%
1M61	405	42	97/109	7.2e-7	100%	2.6e-4	100%
1O4C	274	21	95/104	2.1e-4	100%	4.5e-3	100%
1R6J	270	34	74/82	9.8e-6	100%	4.6e-3	100%
2BYG	426	28	59/97	1.4e-5	100%	7.1e-3	100%

Table 4.4 – Tests with 10 designed positions

rotamers[a]	length[b]	Protein	CFN[c]	Heur.[d]	MC	REMC
2991	108(17)	1A81 3	gmec	0.001	0.1595	
		1A81 4	gmec	0.	0.0317	
		1A81 5	gmec	0.	0.0563	
2520	58(8)	1ABO 1	gmec	0.0675	0.9054	0.8041
		1ABO 4	gmec	0.	0.0128	
2957	98(10)	1BM2 1	gmec	0.	0.0950	
		1BM2 5	gmec	0.	0.1082	
2508	57(8)	1CKA 5	gmec	0.2859	3.2525	0.
2819	91(15)	1G9O 3	gmec	0.1366	0.1366	
		1G9O 5	gmec	0.	3.9599	0.
2957	109(21)	1M61 1	gmec	0.	0.0776	
		1M61 2	gmec	3.5105	4.5062	0.3215
		1M61 5	gmec	0.	0.0432	
3037	104(8)	1O4C 1	gmec	0.	0.1121	
		1O4C 2	gmec	0.	0.1046	
		1O4C 3	gmec	0.	0.1519	
		1O4C 4	gmec	0.	0.1545	
		1O4C 5	gmec	0.	0.1753	
2773	82(10)	1R6J 1	gmec	0.	2.4022	0.3986
		1R6J 2	gmec	0.	1.0398	0.3049
		1R6J 3	gmec	0.	0.0106	
		1R6J 5	gmec	0.	0.0162	
2888	97(15)	2BYG 1	5.7485	0.	0.0337	
		2BYG 3	gmec	0.	0.0833	
		2BYG 4	gmec	0.	0.2149	

[a]Total number of rotamers available to the system. Each designed position can explore 206 rotamers ; the others explore about 10 rotamers each. [b]Total protein length (number of Gly+Pro in parentheses). [c]gmec indicates the GMEC was successfully identified. [d]For all four exploration methods and each test, we report the difference between the best energy obtained and the overall best energy (the best over all methods, which may or not be the GMEC). 10-position tests where all four methods found the GMEC are not listed.

Table 4.5 – Tests with 20 and 30 designed positions

	20 positions					30 positions			
Protein	CFN	Heur.	MC	REMC	mutations[a]	CFN	Heur.	MC	REMC
1A81 1	gmec*	0.	0.3275	0.3851	0	1.2074	0.	0.6353	
1A81 2	gmec*	0.1705	2.4355	1.0069	3	2.5520	0.	0.0578	
1A81 3	gmec	0.	0.4640	0.6186	0	43.5263	0.	2.4996	1.2025
1A81 4	gmec	0.3878	0.5748	0.6991	4	5.1300	0.	0.0305	
1A81 5	gmec	0.0068	0.5088	0.1541	4	3.2417	0.	1.9586	0.5791
1ABO 1	gmec	0.1205	1.1159	0.2153	2	44.5504	0.	0.	
1ABO 2	13.8563	0.	0.	0.	8	12.7303	0.	0.	
1ABO 3	1.2190	0.	0.	0.	9	9.3870	0.	0.2630	
1ABO 4	1.9940	0.	0.0076	0.	5	10.7691	0.	0.	
1ABO 5	3.5418	0.	0.9483	0.9483	9	4.3907	0.	0.	
1BM2 1	gmec	0.	0.0619	0.1584	0	22.5876	0.	1.7290	1.6013
1BM2 2	7.5304	0.	0.0725	0.0143	8	22.1386	0.	1.9856	1.5876
1BM2 3	gmec	0.0229	0.4762	0.2897	0	22.5410	0.	1.9990	1.1541
1BM2 4	0.1186	0.	2.5883	0.0789	2	15.2639	0.	2.2127	2.3854
1BM2 5	gmec	0.2396	0.3746	0.3746	3	15.9890	0.	2.8354	1.1937
1CKA 1	gmec*	0.	0.	0.	0	6.2700	0.	0.	
1CKA 2	gmec	0.	0.	0.	0	2.0995	0.	0.	
1CKA 3	gmec	0.	0.	0.	0	47.0217	0.	0.	
1CKA 4	4.3122	0.	0.	0.	4	44.0830	0.	0.	
1CKA 5	4.2849	0.	0.	0.	3	8.8608	0.	0.	
1G9O 1	2.0574	0.	1.2525	1.2525	5	2.0816	0.	1.5942	0.
1G9O 2	3.2106	0.	0.2177	0.1915	1	0.3270	0.	0.3126	
1G9O 3	1.9008	0.	0.4417	0.1019	1	17.7150	0.	1.5667	1.5667
1G9O 4	0.5030	0.	0.3855	0.1455	5	2.9758	0.	1.4284	1.6202
1G9O 5	0.4298	0.	0.1495	0.5114	5	0.	1.6890	7.6985	2.3857
1M61 1	gmec	0.	0.	0.	0	14.4935	0.0097	0.	0.
1M61 2	gmec	0.	0.	0.	0	5.0899	0.	1.8749	0.008
1M61 3	gmec	0.	0.	0.	0	3.5795	0.	0.0154	
1M61 4	gmec	0.	0.	0.	0	16.1511	0.	0.	
1M61 5	gmec	0.	0.2521	0.1345	0	23.0927	0.	0.	
1O4C 1	0.	0.3465	0.0690	0.0587	6	14.9064	0.	0.3435	
1O4C 2	6.4214	0.	0.1963	0.3175	4	58.1558	0.	0.0795	
1O4C 3	gmec	0.	0.3461	0.0997	0	9.9221	0.	0.1789	
1O4C 4	gmec	0.	0.3640	0.1382	0	5.7790	0.	0.0423	
1O4C 5	0.	0.	0.1131	0.2206	0	9.9221	0.	0.1789	
1R6J 1	gmec	0.	0.2604	0.2002	0	gmec*	0.	0.0246	
1R6J 2	gmec	0.	0.0071	0.0183	0	14.9800	0.	0.0957	
1R6J 3	gmec	0.	0.0537	0.0732	0	0.	0.	0.0440	
1R6J 4	gmec	0.	0.0639	0.0601	0	0.	0.	0.0957	
1R6J 5	gmec	0.	0.0735	0.0244	0	0.	0.7036	1.8823	0.0781
2BYG 1	gmec	0.	3.1878	0.0257	0	17.9752	0.	0.1592	
2BYG 2	gmec	0.	0.0524	0.0831	0	0.3832	0.	0.1502	
2BYG 3	gmec*	0.	1.3564	0.0826	0	0.1442	0.	0.1593	
2BYG 4	gmec	0.	0.1968	0.6022	0	0.	0.0958	0.0050	
2BYG 5	1.8604	0.	0.0933	0.0386	2	0.5003	0.	0.6876	

Format as in Table 4.4. gmec* indicates the more aggressive protocol. [a]Between CFN/Heur.

Table 4.6 – Designed and Pfam sequence entropies

Protein	Top 10,000 structures	Top 10,000 sequences	Pfam seed	Pfam full
1ABO	1.36	1.58	2.79	3.01
1CKA	1.20	1.41	2.84	3.03
1R6J	1.33	1.48	3.11	3.66
1G9O	1.21	1.53	3.29	3.81
2BYG	1.57	1.63	3.31	3.67
1BM2	1.08	1.26	2.90	3.50
1O4C	1.36	1.68	2.94	3.47
1M61	1.31	1.41	2.91	3.51
1A81	1.13	1.29	2.91	3.51

The entropies are exponentiated, then averaged over all positions. The designed entropies correspond to REMC runs where all positions are designed (except Gly/Pro).

Figure captions

1. Sequence logos for the core region of two designed proteins : 1CKA (SH3) and 2BYG (PDZ). The low energy CPD sequences are compared to the sequences of the full Pfam collection of experimental sequences. Positions shown correspond to the hydrophobic core of each protein ; residue numbers are indicated (PDB numbering).
2. Histogram of Blosum40 similarity scores to Pfam sequences for the core region of two designed proteins : 1ABO (SH3) and 1BM2 (SH2). The similarity between Pfam sequences is also shown, considering either the Pfam seed alignment or the much larger full alignment.
3. Run times for different test calculations and search methods. CPU times per core are shown ; only the REMC calculations use multiple cores (OpenMP parallelization). CFN results are only given for 20 designed positions or fewer. REMC was only done for the larger calculations. Results are shown for a large, representative subset of the test calculations. A few CFN calculations were allowed to run beyond the 24h CPU limit. For clarity, the average Heur and REMC values are included as dashed lines.
4. Comparison between selected heuristic, MC, and REMC protocols for whole protein design. The best energy obtained with each protocol is shown for the nine test proteins. Zero represents the overall best energy for each protein. The mean "error" is indicated on the right for each protocol (the mean difference from the overall best energy). The protocols are a heuristic method (Heur), an MC method, two REMC protocols with 4 walkers (REMC a, b), and three REMC protocols with 8 walkers (yellow lines and dots, REMC c, d, e). Details of the protocols are in Table 4.1 ; each curve is labelled according to the protocol name.
5. Lowest energies obtained with the different exploration methods for 5-, 10-, 20-, and 30-position design. Differences with respect to the GMEC or the overall best energy are shown, excluding the smallest values (less than 0.4 kcal/mol above the best energy). The color of each point indicates the nature of the test ; its shape indicates which method gave the best energy. Two examples are highlighted by arrows : the black arrow shows the heuristic result for a 10-position test where the best energy was given by CFN ; the grey arrow shows the MC result for a 20-position test where the best energy was given by the heuristic.
6. Sequence entropy $S(E)$ and number of states $N(E)$ within a given energy range E above the GMEC for the 1CKA SH3 domain. All positions were allowed to vary except Gly/Pro. The entropy (large dots) is a single position sequence entropy, Eq.

(4.13), averaged over all the variable positions. CFN results (black) are based on a complete enumeration of all states within the energy range (at most E kcal/mol above the GMEC). REMC results (grey) are based on the states sampled by all 8 walkers during a single trajectory. The number of states (small dots) corresponds to all the different combinations of sequences and rotamers.

Figure 4.1 – Sequence logos for the core region of two designed proteins : 1CKA (SH3) and 2BYG (PDZ). The low energy CPD sequences are compared to the sequences of the full Pfam collection of experimental sequences. Positions shown correspond to the hydrophobic core of each protein ; residue numbers are indicated (PDB numbering).

Figure 4.2 – Histogram of Blosum40 similarity scores to Pfam sequences for the core region of two designed proteins : 1ABO (SH3) and 1BM2 (SH2). The similarity between Pfam sequences is also shown, considering either the Pfam seed alignment or the much larger full alignment.

Figure 4.3 – Run times for different test calculations and search methods. CPU times per core are shown ; only the REMC calculations use multiple cores (OpenMP parallelization). CFN results are only given for 20 designed positions or fewer. REMC was only done for the larger calculations. Results are shown for a large, representative subset of the test calculations. A few CFN calculations were allowed to run beyond the 24h CPU limit. For clarity, the average Heur and REMC values are included as dashed lines.

Figure 4.4 – Comparison between selected heuristic, MC, and REMC protocols for whole protein design. The best energy obtained with each protocol is shown for the nine test proteins. Zero represents the overall best energy for each protein. The mean “error” is indicated on the right for each protocol (the mean difference from the overall best energy). The protocols are a heuristic method (Heur), an MC method, two REMC protocols with 4 walkers (REMC a, b), and three REMC protocols with 8 walkers (yellow lines and dots, REMC c, d, e). Details of the protocols are in Table 4.1 ; each curve is labelled according to the protocol name.

Figure 4.5 – Lowest energies obtained with the different exploration methods for 5-, 10-, 20-, and 30-position design. Differences with respect to the GMEC or the overall best energy are shown, excluding the smallest values (less than 0.4 kcal/mol above the best energy). The color of each point indicates the nature of the test ; its shape indicates which method gave the best energy. Two examples are highlighted by arrows : the black arrow shows the heuristic result for a 10-position test where the best energy was given by CFN ; the grey arrow shows the MC result for a 20-position test where the best energy was given by the heuristic.

Figure 4.6 – Sequence entropy $S(E)$ and number of states $N(E)$ within a given energy range E above the GMEC for the 1CKA SH3 domain. All positions were allowed to vary except Gly/Pro. The entropy (large dots) is a single position sequence entropy, Eq. (4.13), averaged over all the variable positions. CFN results (black) are based on a complete enumeration of all states within the energy range (at most E kcal/mol above the GMEC). REMC results (grey) are based on the states sampled by all 8 walkers during a single trajectory. The number of states (small dots) corresponds to all the different combinations of sequences and rotamers.

Chapitre 5

PDZ

5.1 Introduction

Nous cherchons maintenant à évaluer performances de notre modèle CPD sur un ensemble de protéines. Les domaines PDZ ("Postynaptic density-95 / Discs large / Zonula occludens-1") sont de petits domaines globulaires qui établissent des réseaux d'interactions entre protéines dans la cellule(1-6). Ils forment des interactions spécifiques avec des protéines cibles, généralement en reconnaissant quelques acides aminés à l'extrémité C-terminale. En raison de leur importance biologique, les domaines PDZ et leur interaction avec les protéines cibles ont été largement étudiées et utilisées en conception in sillico. Des ligands ont été conçus pour moduler l'activité de domaines PDZ impliqués dans diverses pathologie (7-9).des domaines PDZ des ligands PDZ redessinés ont été utilisés pour élucider les principes du repliement des protéines et de l'évolution (10-13).Et ces domaines avec leurs ligands peptidiques fournissent des "benchmarks" pour tester les méthodes informatiques elles-mêmes (14-16).

A partir d'une sélection de domaines PDZ,nous optimisons les énergies de référence E/ref en utilisant un formalisme du maximum de vraisemblance. La performance du modèle est testée en générant des séquences par proteus pour chaque protéine de la sélection. Pour cela, nous utilisons les résultats précédents, en particulier ceux de la section (12), pour définir les valeurs des paramètres de l'optimisation du Monte-Carlo. Nous confrontons nos résultats à ceux de la fonction d'énergie de Rosetta , fonction, plus empirique que la notre, qui a connue le plus de succès(29-31). Elle comprend un terme de répulsion de Lennard-Jones, un terme Coulomb, un terme de liaison hydrogène,un terme de solvatation Lazaridis-Karplus (32) et des énergies de référence d'état dépliées. Il a un Un grand nombre de paramètres spécifiquement optimisés pour CPD, qui offrent des performances optimales, mais une interprétation physique moins transparente que Proteus qui lui fournit la capacité de calculer les énergies libres formellement.

La production de nos séquences calculées a été effectuée par des simulations Monte Carlo où toutes les positions de la chaîne polypeptidique ont été autorisées à muter librement, excepté celles occupées par une glycine ou une proline qui conservent leur type d'acide aminé. Ce qui nous procurent des milliers de variantes pour chaque domaine étudié. Nos tests comprennent une validation croisée où les énergies de référence optimisées sur un sous-ensemble de notre sélection sont utilisés sur un entre sous-ensemble de cette même sélection. Nous, réalisons également,une série de simulations Monte Carlo de deux domaines PDZ où le potentiel chimique hydrophobe des types d'acides aminés est progressivement augmenté, polarisant artificiellement la composition de la protéine. Comme Le biais hydrophobe augmente, les acides aminés hydrophobes envahissent progressivement la protéine De l'intérieur, formant un noyau hydrophobe devenu plus grand que le naturel. La propension de chaque position du noyau à devenir hydrophobe à un niveau de biais plus ou moins élevé peut être considéré comme un indice d'hydrophobicité déterminé en fonction de la structure qui nous renseigne sur la propension du coeur à supporter des mutations.

5.2 Le modèle d'état déplié

5.2.1 Le maximum de vraisemblance des énergies de référence

L'énergie utilisée est ici est l'énergie de pliage de la protéine, c'est-à-dire la différence entre Ses énergies d'état pliées et dépliées (33).Un mouvement élémentaire possible est une "mutation", Nous modifions le type de chaîne latérale t $\rightarrow t'$à$unepositionchoisieidanslaprotéinepliée,enassig$ $E^u = \sum_i E^r(t_i)$

Avec i la position dans la séquence et t$_i letypeeni.Lasommesefaitdoncsurtouslesacidesaminésde S.$

Les grandeurs $E^r(t) \equiv E^r_t$ sont appelées "énergies de référence".Ils peuvent être considérés comme des potentiels chimiques effectifs de chaque type d'acide aminé. Le changement d'énergie de repliement d'une mutation a donc la forme :

$$=^f -^u = (E^f(...t'_i,r'_i...) - E^f(...t_i,r_i...)) - (E^r(t'_i) - E^r(t_i))$$

avec $^f and^u leschangementsd'énergiedansl'étatpliéetdéplié,respectivement.Lesénergiesderéférencesontde$

Plus précisément, nous les choisirons telles qu'elles maximisent la probabilité des Séquences cibles. C'est à dire, nous celles qui sont le plus vraisemblable étant donnée l'observation des séquences cibles.

Soit S une séquence particulière. Sa probabilité de Boltzmann est :

$$p(S) = \frac{1}{Z} exp(-\beta_S),$$

où $_S = G_S^f - E_S^u estl'énergielibrederepliementdeS, G_S^f estdel'étatreplié, \beta = \frac{1}{kT} estlatempératureinvers$
$\sum_i E^r(t_i) - G_S^f - kTlnZ = \sum_t n_S(t) E_t^r - G_S^f - kTlnZ$,où la somme à droite se fait sur l'ensemble des types d'acides aminés et $n_S(t) estlenombred'acideaminésdetypetdansS$.

Nous considérons maintenant un ensemble S de N séquences cibles $\boldsymbol{S}$; On appelle $\boldsymbol{L}$ la probabilité d'observer l'ensemble entier. $\boldsymbol{L}$ est fonction des paramètres du modèle E_t^r.$Commenousvoulonslemaximumde \boldsymbol{L} surles E_t^r, nousseréfèreà \boldsymbol{L} commeleurvraisemblance$.

Nous avons :

$$kTln\boldsymbol{L} = \sum_S \sum_i n_S(t) E^r(t) - \sum_S G_S^f - NkTlnZ = \sum_t n_S(t) E_t^r - \sum_S G_S^f - NkTlnZ,$$

avec N(t) le nombre d'acide aminé de type t dans l'ensemble $\boldsymbol{S}$.Le facteur de normalisation Z (ou fonction de partition) est la somme sur l'ensemble les séquences possibles $\boldsymbol{R}$:

$$Z = \sum_S exp(-\beta_{\boldsymbol{R}}) = \sum_{\boldsymbol{R}} exp(-\beta_{\boldsymbol{R}}^f) \prod_t exp(_{\boldsymbol{R}}(t) E_t^r)$$

Pour maximiser $\boldsymbol{L}$, nous considerons la dérivé de Z selon chacunes des E_t : $_{\frac{r}{t}} = \frac{\sum_{\boldsymbol{R}} n_{\boldsymbol{R}}(t) exp() - \beta_{\boldsymbol{R}}}{}$

$\mathbf{R}$exp(-$\beta_{\boldsymbol{R}}$) = $(t)\rangle$.La quantité à droite est la moyenne de Boltzmann du nombre n(t) des acides aminés t sur toutes les séquences possibles. En pratique, c'est la population moyenne de t que nous voudrions obtenir dans une longue simulation Monte Carlo.

Pour que ln $\boldsymbol{L}$ soit maximal il faut que ses dérivées par rapport à l'E_t^r soient nulles.

$$\frac{1}{N} \frac{\partial}{\frac{r}{t}} ln\boldsymbol{L} = \frac{1}{N} \sum_S n_S(t) - \langle (t) \rangle = \frac{N(t)}{N} - (t)\rangle$$

et donc $\boldsymbol{L} maximum \Rightarrow \frac{N(t)}{N} = \langle n(t) \rangle, \forall t \in aa$

Ainsi, pour maximiser L, nous devrions choisir E_t^r telle qu'une longue simulation donne les mêmes fréquences d'acides aminés que l'ensemble cible.

5.2.2 Recherche du maximum de vraisemblance

Nous utilisons trois méthodes pour approcher les valeurs E_t^r.

1. La première consiste à avancer dans la direction du gradient de ln($\boldsymbol{L}$) en utilisant la règle itérative suivante (40) :

$$E_t^r(n+1) = E_t^r(n) + \alpha \frac{\partial}{r_t} ln(\boldsymbol{L}) = E_t^r(n) + (n_t^{exp} - \langle n(t) \rangle_n)$$

avec $\alpha une constante$,$\mathrm{n}_t^{exp} = \frac{N(t)}{N}$ la population moyenne d'acide aminé de type t dans l'ensemble ciblé, $\langle\rangle_n indique une moyenne sur une simulation effectuée en utilisant les énergies de référ$

2. La deuxième méthode est une variante de la première dans laquelle le n'est pas constant, mais ajusté au cours de la simulation de la façon suivante. La règle (11) est utilisées trois fois avec trois valeurs differentes et constantes pour le ceci avec un jeu d'énergie de références identiques. une interpolation parabolique est effectuée sur les trois valeurs de la fonction proxy obtenues, le minimum de la parabole est calculée et est utilisée comme pour le cycle suivant, en terme duquel les énergies sont mises à jours.
3. La troisième méthode, utilisée précédement (26,27), utilise une règle de mise à jour logarithmique :
avec kT l'énergie thermique, fixée empiriquement à 0,5 kcal/mol. Nous l'appelons la méthode logarithmique. Dans les dernières itérations, certaines valeurs ont tendance à converger lentement,avec des oscillations. Par conséquent, une règle modifié où une énergie au cycle n et l'énergie au cycle n-1 sont moyenenée avec un poids respectifs de 2/3 et 1/3.

Chaque itération, dans la suite, ont étés effectué avec 500 000 000 de pas par réplique de REMC.

5.3 Méthodes de calcul

5.3.1 Fonction énergétique efficace pour l'état replié

La matrice énergétique a été calculée avec la fonction d'énergie efficace suivante pour État plié :

$$E = E_{bonds} + E_{angles} + E_{dihe} + E_{impr} + E_{vdw} + E_{Coul} + E_{solv}$$

Les six premiers termes de l'équation (13) représentent l'énergie interne de la protéine. Ils sont tirés de La fonction d'énergie empirique Amber ff99SB (42), légèrement modifiée pour le CPD.

Les charges du backbone ont été remplacées par un ensemble unifié, obtenu en faisant la moyenne sur l'ensemble des types d'acides aminés et ajuster légèrement pour rendre la par-

tie backbone de chaque acide aminé neutre (43). Le dernier terme à droite de l'équation (13), E_{solv},*représentelacontributiondusolvant.Nousavonsutiliséunmodèledesolvantimplicite"Generalize* *SurfaceArea"ouGBSA*(44) : $E_{solv} = D_G B + E_{surf} = \frac{1}{2}(\frac{1}{\epsilon_W} - \frac{1}{\epsilon_P}) \sum_{ij} q_i q_j (r_{ij}^2 + b_i b_j exp[-\frac{r_{ij}^2}{4 b_i b_j}])^{-\frac{1}{2}} +$ $\sum_i \sigma_i A_i$Ici, ϵ_W et ϵ_P sont les constantes diélectriques du solvant et de la protéine ; r_{ij} est la distance entre les atomes i, j et b_i est le "rayon de solvatation" de l'atome i (44,45). A_i est la surface exposé accessible au solvant de l'atome i.σ_i

est un paramètre qui représente la préférence de chaque atome à être exposé ou caché du solvant. Les atomes du soluté sont divisés en quatre groupes avec pour chacun une valeur $\sigma_i spécifiqueencal/mol/Å^2$:

non polaire -5

aromatique -40

polaire -80

ionique -100

On attribue aux atomes d'hydrogème un coefficient de surface de 0. Les surface sont calculées par l'algorithme de Lee et Richards (46), qui est implementé dans le programme XPLOR, en utilisant un rayon de «probe radius» de 1,5 Å. Les simulations MC utlisent une constante dielectrique = 4 ou 8.

Dans le terme énergétique GB, le rayon de solvatation atomique b_i approxime la distance de i à la surface de la protéine et est une fonction des coordonnées de tous les atomes de protéines. La forme b_i correspond à une variante GB que nous appelons GB/HCT, d'après ses auteurs (44),avec les paramètres du modèle optimisées pour une utilisation avec le champ de force Amber (45).Comme b_i dépend des coordonnés de tous les atomes du soluté (44), une approxamiation supplémentaire est nécessaire pour rendre le terme énergitique GB additif par paire et pour rendre la matrice d'énergie définissable (27,28). Nous utilisons une approximation NEA ("Nativce Environment Approxition"), dans laquelle le rayon de solvatation b_i de chaque groupe (backbone, chaîne laterale ou ligand) est calculé à l'avance , le reste du système étant fixé à sa séquence et sa conformation native. La contribution de l'énergie de surface E_{surf}*n'estpasnonplusadditifparpair,cardanslastructuredelaprotéine,lasurfaceenfouieparunechaî* *evaluerlasurfaceenfuie,unfacteurestappliquéauxzonesdecontactdeschaîneslatéralesimpliquées.D*

5.3.2 Les énergies de référence de l'état déplié

Dans le modèle CPD, l'énergie de l'état dépliée dépend de la composition de la séquence par l'ensemble des énergies de référence E_t^r (équation eqrefEref). Ici, les énergies de référence

ont été attribuées en fonction des types d'acides aminés t, mais aussi de la position de chaque acide aminé dans la structure repliée à travers son caractère enfoui ou exposé au solvant. Ainsi, pour un type donné (Ala, par exemple), il y a deux valeurs distinctes de R_t^r , une enfuie et une exposée. Cette approche se justifie par trois éléments. Tout d'abord, nous supposons que la structure résiduelle est présente dans l' état déplié, de sorte que les acides aminés conservent en partie leur caractère enfui/exposé. Deuxièmement, nous supposons que le modèle d'état déplié compense de manière systématique des erreurs dans la fonction d'énergie de l'état plié, de sorte que la structure pliée contribue indirectement aux énergies de référence. Troisièmement, cette stratégie rend le modèle moins sensible aux variations de la longueur des boucles de surface et au rassio de résidus de surface sur enterrés, qui peut varier considérablement selon les homologues (voir plus bas).

Par conséquent, le modèle devrait être transférable dans une famille de protéines. Distinguer les positions enfouies / exposées double le nombre de paramètres E_t^r à ajuster. Inversement, pour réduire le nombre de paramètres, nous groupons les acides aminés en classes homologues voir table (t1).Dans chaque classe c , et pour chaque type de position (enfoui ou exposé), les énergies de référence ont la forme

$$E_t^r = E_c^r + {}_t^r$$

avec E_c^r est un paramètre ajustable,tandis que ${}_t^r$ est une constante, calculée comme la différence d'énergie de mécanique moléculaire entre les types d'acides aminés de classe c,supposé en conformation dépliée où chaque acide aminé interagit uniquement avec lui-même et avec le solvant.

Plus précisément, nous effectuons des simulations MC d'un peptide étendu (le peptide Syndecan1 ; voir Ci-dessous) et calculons les énergies moyennes pour chaque type d'acide aminé à chaque position peptidique (à l'exclusion des positions terminales). Nous prenons les différences entre les types d'acides aminés et les moyennons sur les positions peptidiques.

Pendant, la maximisation de la vraisemblance, E_c est optimisé tandis que ${}_t est fixe. Pour optimiser les vale$
$)avec des fréquences cibles correspondent aux fréquences expérimentales soient des classes d'acides aminés, s$

5.4 Séquences expérimentales et modèles structurels

5.5 L'ensemble des protéines PDZ

Nous sélectionnons huit protéines de la famille PDZ dont les structures cristalographiques sont connues, avec les trois présentes dans l'ensemble étudié au chapitre précédent :

Groupe	acides aminés	propriétés
1	Ala,Cys,Thr	petit
2	Ser	
3	Glu,Asp	chargé négativement
4	Gln,Asn	polaire
5	Ile,Leu,Val	apolaire
6	Met	non polaire
7	Hip,Hid,Hie	chargé positivement
8	Arg	
9	Lys	
10	Phe,Trp	aromatique
11	Tyr	
12	Gly,Pro	non mutable

Table 5.1 – Les groupes d'acides aminés utilisés pour l'optimisation des énergies de référence.

1G9O,1R6J et 2BYG, aux quelles sont ajoutés 1IHJ, 1N7E, 3K82 et Cask , Tiam1 représenté par .... et ... dans la base de données PDB. Cela constitue un ensemble où le nombre de positions actives , c'est à dire les postions qui vont être muté , est du même ordre pour chaque séquence d'acide aminé des protéines (voir le tableau 5.2).

nom	Code PDB	résidus	nombre de positions actives
NHREF	1G9O	9-99	76
INAD	1IHJ	13-105	82
GRIP	1N7E	668-761	79
Syntenin	1R6J	193-273	72
DLG2	2BYG	186-282	82
PSD95	3K82	305-402	80
Cask	1KWA	487-568	74
Tiam1	4GVD	838-930	84

Table 5.2 – La sélection de domaines protéiques PDZ

Alignements Blast croisés Pour caractériser les homologies dans cet ensemble, une série de requête blast est effectuée sur chaque paire de séquences en utilisant le programme blastp avec les options comme indiqué en (45).Il apparaît que 1R6J et TiAM1 sont atypiques dans l'ensemble avec, aucun homologue avec une E-value inférieure à 1e-7 et

plusieurs E-value supérieur à 10. 3K82 est la protéine plus consensuelle , ayant d'une part une homologie avec toutes les autres à au plus 6e-04 , et d'autre part ayant 4 homologues à moins de 2e-10, pour pourcentage d'identité compris entre 30 et 46.Globalement, il n'y a que peu d'homologies, la plus forte n'étant que de 3e-15 entre 3K82 et 2BYG pour un pourcentage d'identité de 37. Les détails sont dans le tableau 5.3.

Protein	1G9O	1IHJ	1N7E	1R6J	2BYG	3K82	CASK	TIAM1
1G9O	2e-66 (100)	5e-10 (40)	0.002 (25)	3e-07 (25)	2e-11 (35)	1e-12 (30)	5e-05 (25)	9e-07(35)
1IHJ	5e-10 (40)	3e-68 (100)	2e-07 (27)	[18]	2e-08 (27)	9e-14 (46)	4e-06 (35)	[16]
1N7E	0.002 (25)	2e-07 (27)	3e-67 (100)	[21]	3e-14 (36)	2e-10 (37)	9e-12 (30)	5e-05 (35)
1R6J	3e-07 (25)	[18]	[21]	1e-59 (100)	[17]	1e-06 (32)	0.007 (32)	[18]
2BYG	2e-11 (35)	2e-08 (27)	3e-14 (37)	[17]	7e-71 (100)	3e-15 (37)	2e-07 (28)	5e-05 (41)
3K82	1e-12 (30)	9e-14 (46)	2e-10 (36)	1e-06 (32)	3e-15 (37)	4e-70 (100)	1e-07 (27)	6e-04(33)
Cask	5e-05 (25)	4e-06 (35)	9e-12 (30)	0.007 (32)	2e-07 (28)	1e-07 (27)	7e-61 (100)	5e-04(33)
Tiam1	9e-07 (35)	[16]	5e-05 (35)	[18]	5e-05 (41)	6e-04 (33)	5e-04 (33)	1e-68 (100)

Table 5.3 – E-value et pourcentage d'identité des alignements Blast native versus native pour nos séquences PDZ.S'il n'y a pas de touche avec une E-value inférieure à 10,[] donne le pourcentage d'identité du couple dans l'alignement des 6 séquences sauvages.

similarité des homologues Pour définir les fréquences d'acide aminés cibles pour maximiser nos vraisemblances,nous sélectionnons un ensemble de séquences homologues pour chacunes de nos 8 protéines. Pour cela, nous effectuons des recherches blast avec comme requête la séquence extraites du fichier PDB sur la base de données "siwwprot + trEmBL" d'Uniprot avec la matrice BLOSUM62 sans l'option « filtre » et avec l'option « Gapped ». Nous obtenons un premier ensemble pour chaque cas en se limitant aux homologues de bonnes qualité au regard de E-value et du pourcentage d'identité, tout en conservant en même temps une certaine diversité.Cela oblige pour certaines protéine à accepter des E-values plus haute que 1e-40, notamment 1IHJ et 1G9O , respectivement 1e-32 et 1e-10 ,pour avoir un nombre d'homologue suffisant. Ensuite, les redondances les plus flagrantes sont enlevées manuellement. Finalement,les ensembles se composent de 42 à 126 homologues,avec des pourcentages d'identité supérieurs à 66 % excepté pour 1IHJ où il a fallut descendre jusqu'à 38% d'identité.Voir le tableau pour les détails 5.4.

Pour chaque ensemble d'homologues, notons le H, nous calculons la moyenne sur toutes les séquences et toues les prositions pour obtenir des fréquences globales d'acides aminés. Les fréquences sont déterminées séparément pour les positions enfouies et exposées.Notons les $f_t^b(H)$,$f_t^e(H)$, où l'indice t représente un type d'acide aminé et les exposants e et b représentent respectivement aux positions enfuie et exposées. Enfin les ensembles de fréquences moyennes des huit protéines sont eux-mêmes moyennés , ce qui donne deux

protéines	% identité		
1G9O	62	1e-32	67-95
1IHJ	42	1e-10	38-95
1N7E	48	1e-45	84-95
1R6J	85	1e-43	85-95
2BYG	43	1e-41	78-95
3K82	50	1e-46	81-95
Cask	126	7e-28	60-85
Tiam1	50	2e-23	60-85

Table 5.4 – Sélection des homologues.

ensembles cibles distincts de fréquences d'acides aminés f_t^b et f_t^e pour chaque type, et de même pour chaque classe de type.

Protein	1G9O	1IHJ	1N7E	1R6J	2BYG	3K82	CASK	TIAM1
1G9O	326	64	15	15	59	112	49	1
1IHJ	64	221	56	-9	88	107	25	9
1N7E	15	56	378	24	65	87	90	39
1R6J	15	-10	24	311	-26	22	42	-18
2BYG	59	88	65	-26	325	110	24	22
3K82	112	107	87	22	110	325	66	21
Cask	49	25	90	42	23	66	308	37
Tiam1	1	10	39	-18	22	21	37	371

Table 5.5 – Similarité des séquences expérimentales homologues, pour les 8 protéines PDZ.

similarité des homologues

Pour réaliser les calculs Monte Carlo, les structures ont été préparés et les matrices d' énergie calculées à l'aide d'une procédure décrites précédement (15,50).Deux sgments manquants dans le domaine Tiam1 (résidus 851-854 et 868-869) ont été construits en utilisant le programme Modeller (51). Le ligand peptidique a été retiré de la structure PDB avant de calculer la matrice d'énergie.Pour chaque paire d'acide aminés, l'énergie d'interaction a été obtenue après 15 pas de minimisation de l'énergie, avec le backbone fixé et seulement les interactions de la paire entre les autres chaînes et le backbone. Cette courte minimisation simplifie l 'approximation discret. Les rotamères de chaînes latérales utilisés sont une version légérement étendue de la librairie de Tuffery et cal (52), qui poséde un total de 254 rotamères (sur l'ensemble des types d'acides aminés).Cette extension

comprends des orientations d'hydrogème supplémentaires pour les groupes OH et SH (48). Cette bibliothèque de rotamères a été choisié pour sa simplicité et parce qu'elle a donné de très bonnes performances dans les tests de placement de chaînes en comparaison au programme spécialisé scwlr4 qui utilise une bibliothèque beaucoup plus grande (53,54).

aa	1G9O	1IHJ	1N7E	1R6J	2BYG	3K82	cask	tiam1
ALA	5.8	10.4	14.5	8.3	12.1	4.6	4.6	7.1
CYS	3.0	1.6	0.1	2.6	0.2	0.4	3.0	0.0
THR	3.0	1.4	6.1	8.9	6.7	2.5	4.4	3.0
SER	4.2	8.5	3.2	7.2	1.8	7.1	4.4	4.8
GLU	7.4	1.4	0.0	0.3	0.1	6.3	6.3	5.9
ASP	6.3	3.1	5.9	0.2	8.0	2.4	3.9	3.0
ASN	2.9	0.3	2.9	3.3	3.9	2.4	0.7	2.9
GLN	3.2	3.0	0.0	0.7	1.1	4.7	1.4	0.1
ILE	7.0	22.1	23.4	17.0	13.3	3.3	19.7	11.6
VAL	25.8	16.4	7.9	18.8	18.6	1.8	13.8	13.1
LEU	17.2	13.6	29.9	14.6	18.8	5.3	15.1	25.5
MET	1.2	0.8	0.1	2.6	0.0	0.6	8.4	1.5
HID	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HIE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HIP	0.0	0.7	0.0	3.4	2.6	0.3	1.2	0.1
ARG	2.8	6.5	2.9	0.3	0.2	4.4	0.6	2.9
LYS	0.1	1.8	0.2	5.7	4.4	2.7	7.1	5.8
TRP	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0
PHE	6.4	6.8	0.2	4.2	3.08	4.3	3.9	5.9
TYR	3.4	0.3	2.8	1.1	2.6	5.4	0.0	5.7
GLY	0.1	0.9	0.0	0.5	2.3	1.0	0.0	0.0
PRO	0.1	0.3	0.0	0.0	0.1	0.2	0.6	0.0

Table 5.6 – Compositions en acides aminés des séquences expérimentales homologues aux positions enfouies et actives. pour les 8 protéines.

5.5.1 simulation Monte Carlo

La conception des séquence est réalisé avec Proteus,

D'abord,pour optimiser les énergies de référence, nous faisons des simulations où environ la moitié des position peuvent muter à la fois. Ensuite,les modèles optimisé ont

été testés avec des simulations où tous les positions sauf sauf celles nativement en Gly ou Pro, sont libre de muter. C'est également de cas pour les simulations de la titration hydrophobe.

Dans les deux cas des mutations se sont produites au hasard, soumises uniquement à la fonction d'énergie MMGBSA Qui entraîne la simulation. Les simulations Monte Carlo utilisent des mouvements à une ou deux positions,où les rotamères,les types d'acides aminés ou les deux peuvent changer.Pour les mouvements à deux positions, la deuxième position est choisi parmi celles qui avaient une énergie d'interaction significative avec la première (c'est à dire 10 kcal/mol ou plus). De plus, l'échantillonnage est amélioré par l'échange de réplique (REMC), où plusieurs simulations MC sont exécutées en parallèle, à différenres températures. Des échanges périodiques

5.5.2 Génération de séquence Rosetta

Des simulations Monte Carlo ont également été réalisées à l'aide du programme et de la fonction d'énergie Rosetta (37).Les simulations sont faites en utilisant la version 2015.38.58158 de la suite (librement disponible en ligne), en utilisant la commande :

Fixbb -s Tiam1.pdb -resfile Tiam1.res -nstruct 10000 -ex1 -ex2 -linmem_ig 10

où les options ex1 et ex2 activent une recherche améliorée des rotamères pour les chaînes latérales enfouis. La dernière option correspond au calcul de l'énergie à la volée, et les paramètres par défaut sont utilisées pour les autres options. Les résidus Gly et Pro présents dans la protéine sauvage ne sont pas autorisé à muter, et les positions qui mutent ne peuvent pas le faire en Gly ou Pro (comme dans les simulations Proteus).Des simulations sont exécutées pour chaque domaine PDZ jusqu'à obtenir 10 000 séquences uniques de faible énergie, ce qui correspond à des durées d'éxecution d'environ 5 minutes par séquence sur un seul coeur d'un processeur Intel récent, pour un total de 10 heures par protéines en utilisant 80 coeurs.C'est tout à fait comparable au coût des calculs Proteus , en comptant le temps de caluls de la matrice d'énergie plus celui des simulations Monte Carlo.

5.5.3 Caractérisation de la séquence

Les séquences calculées sont comparées à l' alignement Pfam pour la famille PDZ, en utilisant a matrice Blosum40 et une pénalité d'écart de -6 ;Cette matrice est approproée pour comparer des homologies éloignés (séquences CPD et Pfam ici). Chaque séquence Pfam est également comparée à l'aligmement Pfam, ce qui permet de comparer des séquences calculées et un couple de domaine PDZ naturels. Pour ces comparaisons Pfam/Pfam, si

un domaine de test T fait partie de l'aligmenent , la comparaison T/T n'est pas prise en compte, pour être plus cohérent avec les comparaisons calculées/Pfam.L'alignement Pfam utilisée est "RP55" , composée de 12255 séquences. Les similitudes sont calculées pour d'abord les 14 résidus du coeur et pour 16 résidus de surface définis par leur presque total enfuissemebt ou exposition (voir ? ?) et enfin pour l'ensemble des positions de la protéine.

Les séquences calculées sont soumises à la bibliothèque de modèle de Markov Caché Superfamily (57,58) qui tente de classer les séquences selon la base de donnée structurelle de protéines SCOP (59). La classification étant basée sur la version 1.75 de SCOP et 3.5 des Superfamily(voir plu haut).Le programme hmmscan est exécuté avec un seuil de valeur E de $10^{-10} et un total de 15438 modèles dans la base SCOP.$

Pour comparer la diversité des séquences produites avec la diversité des séquences naturelles, nous utilisons l'entropie par position, à partir de la formule $S_i = -\sum_{j=1}^{6} f_j(i) ln f_j(i)$

avec $f_j(i)$ la fréquence du type de résidu j à la position i, Au lieu de distinguéles 20 types d'acide aminés, nous ut
$LVIMC$,FYW,G,$ASTP$,$EDNQ$etKRH. Cette classification est a été obtenu par une analyste de clusters sur l

5.6 Résultats

5.6.1 Structures et séquences expérimentales

Les structures tridimensionnelles (3D) des quatre domaines PDZ de test sont représentées sur la Fig. 1A. Quatorze résidus de noyau (identifiés visuellement) des différentes structures se superposent bien. Tandis que les boucles et les chaînes terminales affichent de grands écarts. L'hélice $\alpha 2 de Tiam 1 est tournée légèrement vers l'extérieur par rapport aux trois autres structures (70).$
$ment et par ailleurs les identités des paires des séquences. Les écarts se situe entre 1,0 et 2,1 et des identités des séque$

La conservation des séquences dans les quatre domaines PDZ et un sous-ensemble de l'alignent "seed" de Pfam est représenté sur la figure 2. Les 14 positions utilisés pour définir le noyau hydrophobe sont bien conservé conservé dans l'alignement des "seed" de Pfam, mais pas totalement. L'Arg, Lys et Gln apparaissent à certaines positions, puisque dans de petites protéines comme des domaines PDZ, Le longue partie hydrophobe de ces chaînes latérales peut être enfui dans le noyau tout en permettant à la pointe polaire de la chaîne d'être exposé au solvant. Quelques résidus Asp et Glu apparaissent aussi, dans les endroits où l'alignement des séquences peut ne pas très bien refléter la superpositions 3D les chaînes latérales.

5.6.2 optimisation du modèle de l'état déplié

Nous optimisons les énergies de référence $E_t^r pour les six protéines, en utilisant leurs homologues natu$
$classe dépend par construction du_t^r défini dans pour chaque classe, qui on tété calculés avec la mécaniquen$
$huit types possibles (les positions Pro et Gly étant fixées).$

5.6.3 Évaluation de la qualité des séquences obtenues

Tests de reconnaissance de famille Les simulations Proteus utilisent l'algorithme Monte Carlo avec échange de réplique (REMC) avec huit répliques et 750 millions de pas par réplique, avec des énergies thermiques kT qui varient de 0,125 à 3 kcal/mol. Toutes les positions sont autorisés à muter librement dans tout les types d'acides aminés excepté Gly et Pro. Les simulations ont été faites avec la fonction d'énergie MMGBSA, sans aucune introduction de biais vers les séquences naturelles ni aucune limite sur le nombre de mutations. Les 10 000 séquences avec les énergies les plus faibles parmi celles échantillonnées par au moins une des répliques MC sont retenue pour l'analyse. De la même façon, 10 000 séquences produites par Rosetta ont était retenue. Ces séquences sont analysées par les outils de reconnaissance de repliment "Superfamily" (58,71) , voir tableau 4.Avec une constante diélectrique de 8, nous avons obtenu un pourcentage élevé de séquences correctement associées à la famille et superfamille PDZ : 91% pour Tiam1 et 100 pour Cask, avec des "E-values" d'environ
$10^{-3} pour les affectations à la familles. Ces valeurs sont semblables à celles obtenues par Rosetta (90 et 98\%$

Séquences et diversité de séquence Les séquences Tiam1 et Cask calculées par Proteus , par Rosetta et les séquences naturelles sont montrées à la figure 3. Pour les quatorze résidus du noyau et la figure 4 pour les seize résidus de surface (Tiam1 uniquement). Les séquences sont représentées par des logos de séquences. Comme on l'a vu dans de nombreuses études de CPD antérieures (30,72) l'accord avec l'expérience pour les positions du coeur est très bon, alors que l'accord en ce qui concerne les résidus de surface est nettement plus faible. La diversité des séquences naturelles et des séquences calculées est caractérisé par la moyenne sur la séquence de l'exponentielle de l'entropie résiduel (voir Méthodes), ce qui correspond à un nombre moyen de classe de séquence échantillonnées par position. Par exemple, une valeur de 2 à une position particulière indique que les acides aminés de deux des six classes sont présents à cette positions au sein de l'ensemble des séquences analysées. Une valeur moyenne globale de deux indique qu'en moyenne, deux classes d'acides aminés sont présentes à n'importe quelle position dans les séquences analysées. Comme référence l'ensemble Pfam RP55 et 12 255 séquences naturelles a une entropie moyenne de 3,4.Le regroupement des séquences Tiam1 et Cask calculés donne

une entropie de 2,2 avec Rosetta et 2,0 avec Proteus. Ce qui indique que ces deux seules géométries de backbone ne peuvent pas être attendre les mêmes niveaux de diversités que le grand ensemble RP55. Prenant les 10000 séquences Monte Carlo de meilleure énergie échantillonné à température ambiante (au lien des 10 000 de plus basses énergies échantillonnées collectivement par toutes les répliques à toutes les températures) et la mise en commun de Tiam1 et Cask donne une tropie globale plus élevée de 2,9 avec Proteus. Pour Rosetta, l' entropie dans le noyau est seulement légèrement inférieur à la moyenne sur toutes les positions. Pour Proteus, c'est nettement inférieur (1,25). Pour les séquences Pfam-Rp55, cette entropie est de 1,8.

Scores de similarité Blosum La figure 5 montre les scores de similarité Blosum40 entre les séquences calculées et les séquences naturelles. Avec Proteus, pour Tiam1 et Cask, les similitudes globales se chevauchent au pied du somment des scores naturels, sont comparables aux valeurs des séquences Rosetta. Pour les résidus de suface, montrés séparément, la similitude avec les séquences naturelles sont faibles (scores inférieur à zéro), à la fois pour Proteus et Rosetta. Avec $\epsilon_p = 8$,$Proteusdonnepresquelamêmesimilitudemoyennesurtouteslespostions Tiam1 et Ca$

Tests de validation croisée Comme premier test de validation croisée,nous utilisons les énergies de référence optimisées à l'aide des homologues de Tiam1 et Cask à deux entre domaines PDZ : DLG2 et syntenine. Les scores Superfamily sont comparables à ceux obtenus pour Tiam1 et Cask, avec 100% de reconnaissance de la famille (tableau 4). Les séquences calculées avec Rosetta pour la DLG2 et la syntenine ont également obtenues une reconnaissance de la famille dans 100% des cas. Comme validation croisée supplémentaire,nous optimisons les énergies de références en utilisant un ensemble alternatif de domaines PDZ : DLG2, syntenin, PSD95, GRIP, INAD et NHERF. Pour distinguer les variantes du modèles, nous nous reférons a cette nouvelle variante en tant que modèle n=6 (il utilise six domaines PDZ pour la paramétrisation) et le modèle initiale comme modèle n=2. Les énergies de référence nouveau n=6, sont alors utilisées pour produirent des séquence Tiam1 et Cask , qui sont alors soumisses aux tests Superfamily et des calculs de similarité. La performance de Tiam1 sur la super-famille est légèrement dégradée par rapport au précédent modèle n=2.Le score superfamily de Tiam1 diminue de 90,6% à 76,6% pour la reconnaissance de la famille. Les score de Cask est inchangé. Les histogrammes des scores de similarité Blosum montrent que les scores globaux pour Tiam1 et Cask avec n=6 sont très semblables à ceux du modèle n=2, alors que les scores pour les positions centrales sont nettement améliorés. Pour DLG2 et syntenine, nous calculons également les scores de similarité en utilisant à la fois le modèle n=2 et le modèle n=6. Les scores

de similarité avec n=2 sont légèrement plus faibles qu'avec n=6, comme on pouvait le prévoir.Le score global a diminué d'environ 20 points pour la synténine et environ 10 points pour DLG2. dans l'ensemble, les modèles de validation croisée ont légèrement dégradé les performances. Ainsi, pour tout domaine d'intérêt PDZ, Il est préférable d'optimiser les énergies de référence spécifiquement pour ce domaine plutôt que de transférer des valeurs paramétrées en utilisant d'autres domaines PDZ.

Stabilité des séquences calculées dans les simulations de dynamique moléculaire

Un autre test du modèle a été effectué au Laboratoire par Nicolas Panel. Dix séquences tiam1 conçues avec proteus sont soumises à des simulations de dynamique moléculaire (MD) en utilisant un environnement de solvat explicite ; Ces séquences sont obtenues en utilisant Proteus avec $\epsilon_p = 8 ou \epsilon_p = 4. Bienqu'aucunligandpeptidiquenesoitprésentpendantlagénérationdesséq$

1. Les séquences doivent avoir un point isoélectrique non neutre
2. Elles doivent être assignées à la bonne famille SCOP par Superfamily avec de bonnes "E-values"
3. Elles doivent avoir de bons scores de similarité Pfam
4. Elle doivent avoir au plus 15 mutations qui modifie le type d'acide aminé par rapport à la protéine sauvage. C'est à dire une mutation telle que le score BLOSUM62 associé soit inférieur au égal à -2.

   L'application de ces critères ont réduit de l nombre de séquences à 66 pour $\epsilon_p = 8 et 45 pour \epsilon_p = 4. Enoutre, ontétééliminéeslesséquencesayantdeuxmutationsquicréentunecavi$
   $6 ouplus (cequipeutentraînerl'instabilitédesprotéines). Aufinal, sixséquencessontchoisiespour$
   $6 ouseq-1,...,seq-6. Lesséquences 1,2,4, et 5 ontétémodifiéesmanuellementpouréliminerlesrési$
   $856 (leslysinechangéesonalanine), donnantdesséquences 1', 2', 4' et 5'. Lesdixséquencessontprés$
   $values" Blastd'environ 10^{-8} - 10^{-7} (saufuneséquenceavecune" E-value" à 10^{-10}). Lesdixséquer$
   $857). Aucoursdelatrajectoire MD, l'écartdermsparrapportàlastructure MD moyennevariede 1$
   $2) semblestablejusqu'à près de 1000 ns (figure 6B); Lastrctureseq-2 moyenneamontréunraccou$
   $2 MD, l'écartdermsdeseq-2 Desastructure MD moyennevariaitentre 1,3 et 2 Â jusqu'à près de 100$
   $2 connaitunefortefluctuation. Sur 100 nssupplémentairesdesimulation, lafluctuationrégresse$
   $6 et WT sontsuperposésà lafigure 6C etsonttrèssemblableslesunsauxautres, avecunedéviationd$
   $eux. Aucoursdelatrajectoire MD deseq-6, lesécartsparrapportàsastructure MD moyenneaflu$

5.7 Application : Croissance du noyau hydrophobe

Comme application de nos modèles optimisés, nous examinons la possibilité de conception du coeur hydrophobe des domaines PDZ. Chaque domaine PDZ est soumis à une simulation

REMC avec une succession de fonctions énergétiques biaisées qui favorisent de plus en plus les résidus hydrophobes. La première simulation comprend un terme d'énergie de biais $\delta =$ $0{,}4kcal/mol(parposition)quipénaliselestypesd'acidesaminéshydrophobes(I,L,M,V,A,W,FetY).Ladernièr$ $-0.4kcal/mol(parposition)quifavoriselestypeshydrophobes.Lesvaleursd'énergiedebiaisintermédiaire\delta =$ $0{,}2kcal/moletδ = -0{,}2kcal/molsontégalementsimulé.Endiminuantprogressivementlavaleurdubiaisd'én$

Les résultats pour Tiam1 sont présenté à la figure 7. A la plus grande valeur $\delta lecoeurhydrophobedeTiam1es$ $/-12changements),reflétantlebiais.Environ2/3deschangementsseproduisentdansdeélémentsdestructur$

Nous calculons également un paramètre pour décrire le nombre de changement relatif de type d'acide aminé par unité de l'énergie du biais. Ce paramètre est défini comme le nombre de positions de résidu changées de non-polaire à polaire, divisé par le produit de la variation dans l'énergie de polarisation et le nombre moyen N de position non polaires à biais nul. Nous l'appelons la sensibilité hydrophobe $\psi_h.PourledomainePDZTiams1,cecalculdonne$: $\psi_h = \frac{1}{N}frac = 0{,}88changementsparpositionetparkcal/mol.PourCask,lasensibilitéhydrophobeest\psi_h =$ $0{,}71changementsparpositionetparkcal/mol.$

5.8 Discussion

5.8.1 limites du modèle

Nous paramétrons un modèle CPD simple pour la conception de domaine PDZ, adapté aux applications à conception de nombreuses séquences et mis en oeuvre dans le logiciel Proteus. Pour la modélisation de l'état replié, nous utilisons un champ de force protéique de haute qualité. Nous avons testé deux ensembles de paramètres d"énergie de surface atomique $\sigma_ietdeuxmodèlesdesolvant"GBgénéralisé",lemodèleNEAetleexactGB,avecuneconstantediélectrique\epsilon_P =$ $8pourleNEA,et\epsilon_P = 4pourleexactGBquantitésquicaractérisentlemodèledesolvantetsontlesprincipauxpa$

La représentation de l'état déplié utilise un modèle simple caractérisé par une ensemble de potentiels chimiques d'acide aminé empiriques ou énergies de référence. Ces énergies sont choisi par une procédure de maximisation de la probabilité, formulée ici, afin de reproduire la composition d'acides aminés d'homologues naturels soigneusement sélectionnés. L'état déplié utilisé ici est plus raffinée qu'auparavant (76), puisque des valeurs d'énergies de référence distinctes sont utilisés pour les positions d'acides aminés qui sont enfuis ou exposés à l'état replié.

Cette méthode suppose qu'il existe une structure résiduelle à l'état déplié, où certaines positions sont plus enfuis que d'autres. En outre, cela doit rendre le paramétrage plus robuste et moins sensible à la taille et à la structure des homologues naturels utilisés poue définir les compositions d'acide aminés cibles, car les fréquences d'acide aminés des positions exposés et des régions enfouies sont calculées séparément. En principe, cela double le nombre d'énergie de référence à ajuster. Cependant, nous avons réduit ce nombre en introduisant des classes de similarité d"acide aminés, avec une énergie de référence ajustable par classe. Contrainte qui est levé en fin d'optimisation. Lors de l'optimisation de énergie de référence, nous effectuons, des calcules de séquences pour chaque protéines de notre jeu de test (à l'état apo) où une positions sur deux peut muter, soit la moitié des positions (à l'exception de Gly et Pro), avec une simulations distinctes pour chaque moitié. De cette façon, lors de l'optimisation des paramètres, une position mutable est toujours entouré d'un environnement identique au type sauvage au moins sur les deux positions immédiatement voisines sur le squelette. Les calculs de conception des protéines s'appuient sur une méthode d'exploration Monte Carlo avec échange de réplique, puissante et efficace, qui utilise plus d'un demi-miliard de pas par simulation et par réplique, et produit des milliers de séquences dans une seulle simulation. Les valeurs des énergie de référence sont optimisées avec deux choix différents du couple modèle de solvant constante diélectrique des protéines ϵ_p.$Lesperformancesontaméliorés pourlemodèleexactGB.Lemodèleprésenteplusieurslimitation,don$
$4ou8)(78).Cettevaleurdiélectriquesignifiequelastructureprotéique(ycomprissonsquelette)estautori$
$backbonearécemmentétédéveloppéedansProteus,surlabased'uneméthodeMonteCarlohydridequiprés$

Une autre limitation de notre modèle est la nécessité, pour des résultats optimaux, de paramètres les énergies de référence spécifiquement pour un ensemble donné de protéines. Cette étape est bien automatisée et de façon très parallèle. Cependant, cela implique plusieurs choix qui sont partiellement arbitraires. Ceux-ci comprennent le choix d'un ensemble de domaines protéiques pour représenter la protéine ou la famille d'intérêt. Nous devons également choisir un seuil de similarité pour définir les homologues cibles à partir desquels sont calculées les compositions expérimentales d'acides aminés. Ici, nous avons choisi d'utiliser les homologues de chaque membre de la famille, de calculer leurs compositions, puis la moyenne sur les six familles. Cette méthode a correctement fonctionné, mais d'autres choix sont possibles et il faut travailler d'avantage pour pouvoir tirer des conclusions définitives sur ces choix.

Une autre limitation de notre modèle est l'approximation de la position des chaînes latérales en rotamères discrets. qui nécessite une certaine adaptation de la fonction d'énergie

pour éviter les affrontements stériques exagérés. La méthode utilisée ici les t la méthode de minimisation de la paire de résidu décrite précédemment (26,76).

ailleurs ! ! Une cinquième limitation est l'utilisation d'un modèle de solvatation additif par paires (comme dans la plupart des modèles CPD). Plus précisément, l'environnement diélectrique de chaque paire de résidus est supposé ici être celui de la structure native (ce qu'on appelle "Approvisionnement environnemental natif" ou NEA 74,76). Cela conduit à une fonction énergétique qui a la forme d"une somme sur les paires de résidus et peut être pré-calculée et stockée dna une matrice énergétique, qui sert alors de table de consultations pendant les simulations Monte-Carlo. Malgré cette approximation, le modèle a donné de bons résultats pour un grand nombre d' indice acide/base de référence, un problème très sensible au traitement électrostatique 74. Certaines de ces limitations sur supprimées dans le modèle exactGB. En particulier, comme la fonction d'énergie est principalement basée sur la physique, elle a pu être améliorée par implémentions d'un calcul "GB" plus exact. Par ailleurs, il a été mis en place une modèle amélioré pour la solvatation hydrophobe (80), qui est plus rapide et plus précise que notre terme d"énergie de surface actuelle (article préparation).

5.8.2 modèle de test et applications

Les séquences calculées sont largement comparées aux séquences naturelles, à travers des tests de reconnaissance du pli, des calculs de similarité et des calculs d'entropie. Dans les simulations, nous concevons la totalité de la séquence de la protéine, de sirte que toutes les positions (à l'exception de Gly et Pro) peuvent muter librement, soumis seulement à un biais vers la composition moyenne expérimental des acides aminés (à travers les énergies de références). Malgré la quasi-absence de biais expérimenaux ou de contraintes, les séquences obtenues ont une forte similitude globale avec les séquence naturelles de Pfam, mesurées par les scores de similarité Blosum40. Les scores obtenus sont, pour l'essentiel, comparables aux scores de similarité entre les paires de séquences Pfam entre-elles. Ainsi, les séquences conçues avec Proteus ressemblent à des homologues naturels modérément éloignés. La similitude est très forte pour les résidus au coeur de la protéine, comme cela a été observé dans des études CPD précédantes (30,72). En revanche, pour les résidus à la surface de la protéine, les scores de similarité sont proche de zéro, le score obtenu si l'on tirait au hasard les acides aminés a ces positions. Notez que de nombreux résidus de surface sont impliqués dans des interactions fonctionnelles, comme les onze résidus de liaison aux peptides dans les domaines PDZ. Les résidus de surface sont également sélectionnés

selon l'évolution pour éviter l'agrégation ou des adhésion indésirable. La plupart de ces contraintes fonctionnelles ne sont pas explicitement pris en compte dans notre protocole de design (bien que la fonction d'énergie soit indirectement la solubilité des protéines). Malgré les scores de similarité limités pour les régions de surface, la reconnaissance de pli avec l'outil Superfamily appliquée aux meilleures modèles concues est presque parfaite. Les test de reconnaissance de plie antérieurs qui utilisaient une fonction d'énergie plus simple donnaient un taux de reconnaissance de pli inférieur, à environ 85% (pour un ensemble de tests plus large et plus diversifié) et des similitudes inférieures (15,50).De toute évidence, l'utilisarion combinée d'un champ de force protéique amélioré, du solvant "GB" et des énergie de référence spécifiques à la famille conduisent à des séquences calculées sont proche des séquences natives et sans doute meilleures.

les séquences Proteus ont également été comparées aux séquences obtenues avec le logiciel Rosetta , qui a lui-même été testé de manière approfondie. Sur la Base des scores de similarité de Blosum (par rapport aux séquences naturelles dans Pfam) et des tests de reconnaissance du pli, les séquences Proteus et Rosetta semblent être de la même qualité. Cependant, rosetta fait moins de mutations que Proteus ; de sorte que les scores d'identité, par rapport à la protéine de type sauvage correspondante, sont sont environ 6% plus haut chez Rosetta. Ce qui veut dire que Proteus modifie environ 5 positions en plus , en moyenne, par domaine PDZ. Ce nombre peut facilement être réduit en ajoutant à la fonction d'énergie de Proteus un terme d'énergie de biais explicite qui augmente avec le nombre de mutations l'éloignant de la séquence de type sauvage.

Une autre possibilité pour tester nos séquences est d'utiliser les simulation de dynamique moléculaire. Nicolas Panel a testé dix séquences Tiam1 calculées dans des simulation MD assez longues, dans la forme apo, en utisant le même champ de force protéique que dans le modèle CPD (champ de force Amber) et un modèle d'eau explicite. Ces séquences

Conclusion

XXX

Annexe 1 :Liste des positions actives pour chaque test

Nom	S_{Vois}	positions actives
1A81 1	10	10 13 16 84 86
1A81 2	10	20 21 24 27 116
1A81 3	10	35 38 56 105 107
1A81 4	10	44 47 52 65 67
1A81 5	10	82 84 86 87 90
1ABO 1	10	64 66 90 93 100
1ABO 2	10	72 74 80 104 111
1ABO 3	10	79 82 102 111 115
1ABO 4	10	83 86 104 105 106
1ABO 5	10	93 100 102 113 116
1BM2 1	10	101 106 140 141 146
1BM2 2	10	120 128 131 132 135
1BM2 3	10	58 61 127 128 129
1BM2 4	10	74 75 98 100 105
1BM2 5	10	85 87 95 110 128
1CKA 1	10	136 138 158 175 190
1CKA 2	10	149 166 169 171 181
1CKA 3	10	151 153 157 159 172
1CKA 4	10	164 170 172 184 187
1CKA 5	10	172 174 182 186 187
1G9O 1	10	10 13 54 57 92
1G9O 2	10	15 39 42 54 57
1G9O 3	10	24 26 28 39 42
1G9O 4	10	48 53 57 59 88
1G9O 5	10	75 78 79 86 88
1M61 1	10	12 20 23 24 27
1M61 2	10	17 20 24 37 49
1M61 3	10	27 33 51 100 102
1M61 4	10	5 8 10 11 36
1M61 5	10	59 71 84 87 94
1O4C 1	10	20 21 32 34 46
1O4C 2	10	2 71 79 81 82
1O4C 3	10	33 45 63 71 73
1O4C 4	10	43 45 63 71 85
1O4C 5	10	8 33 82 83 86
1R6J 1	10	194 237 239 270 272
1R6J 2	10	199 201 211 218 232
1R6J 3	10	213 218 227 232 238
1R6J 4	10	221 227 232 267 269
1R6J 5	10	241 254 258 267 269
2BYG 1	10	189 191 221 244 246
2BYG 2	10	205 224 239 245 248
2BYG 3	10	232 233 265 272 274
2BYG 4	10	238 240 243 276 278
2BYG 5	10	253 261 264 265 274

Table 7 – Les tests avec cinq positions actives

Nom	S_{Vois}	positions actives
1A81 1	10	13 15 39 41 53 86 89 90 93 103
1A81 2	10	39 41 53 55 64 66 76 89 92 103
1A81 3	10	51 53 64 66 68 74 76 82 88 92
1A81 4	10	76 82 87 88 90 91 92 95 97 99
1A81 5	10	9 10 11 16 41 51 53 66 88 89
1ABO 1	10	64 72 74 79 89 91 101 103 108 111
1ABO 2	10	66 68 80 82 88 90 100 102 104 111
1ABO 3	10	69 70 72 74 80 81 106 113 114 115
1ABO 4	10	71 78 83 84 94 99 101 104 105 106
1ABO 5	10	72 79 82 94 99 102 104 106 111 115
1BM2 1	10	119 120 121 122 123 125 131 134 135 140
1BM2 2	10	125 126 127 129 130 133 134 136 137 147
1BM2 3	10	83 99 101 106 108 135 140 141 146 148
1BM2 4	10	85 95 97 110 118 120 125 128 131 132
1BM2 5	10	99 101 106 139 140 141 142 143 144 146
1CKA 1	10	134 135 160 161 162 173 174 175 176 179
1CKA 2	10	137 139 143 151 153 157 159 172 182 186
1CKA 3	10	138 140 147 149 150 155 166 169 181 188
1CKA 4	10	140 141 153 154 155 157 174 175 184 186
1CKA 5	10	151 153 157 166 168 173 174 176 178 179
1G9O 1	10	10 11 13 14 15 16 53 54 57 92
1G9O 2	10	15 17 24 26 39 42 48 51 53 88
1G9O 3	10	26 28 39 42 48 53 57 59 88 90
1G9O 4	10	34 35 58 60 68 70 74 75 89 91
1G9O 5	10	71 73 74 77 80 81 82 83 84 85
1M61 1	10	10 12 20 23 24 27 35 49 102 104
1M61 2	10	17 20 21 24 37 39 40 47 49 58
1M61 3	10	34 36 46 48 59 61 71 83 84 87
1M61 4	10	5 6 11 36 46 48 61 69 83 84
1M61 5	10	59 61 70 71 75 77 83 86 87 92
1O4C 1	10	31 33 45 47 61 63 73 86 89 100
1O4C 2	10	50 51 52 53 63 72 73 77 85 89
1O4C 3	10	61 62 63 71 72 73 79 85 88 89
1O4C 4	10	73 74 75 76 77 89 92 94 96 101
1O4C 5	10	90 91 93 96 98 99 101 102 103 104
1R6J 1	10	193 194 195 197 199 218 232 236 267 269
1R6J 2	10	199 209 211 213 218 227 232 238 265 267
1R6J 3	10	201 204 205 209 211 218 241 258 265 267
1R6J 4	10	209 211 213 218 227 238 241 258 265 267
1R6J 5	10	238 240 241 242 246 257 258 261 265 267
2BYG 1	10	194 196 203 205 224 233 239 245 274 276
2BYG 2	10	203 205 207 224 227 233 239 243 245 276
2BYG 3	10	206 207 222 245 248 253 256 261 264 265
2BYG 4	10	221 222 245 248 251 253 256 261 264 265
2BYG 5	10	247 248 249 250 251 252 259 262 263 275

Table 8 – Les tests avec dix positions actives

Nom	S_{Vois}	positions actives
1A81 1	1	9 11 12 13 15 16 17 19 20 25 28 39 40 41 42 43 44 45 114 117
1A81 2	1	9 11 12 13 15 16 17 19 20 25 28 39 40 41 42 43 44 45 47 117
1A81 3	1	9 11 12 13 15 16 17 19 19 41 43 48 51 68 74 84 86 109 114 117
1A81 4	1	12 13 15 16 17 19 20 25 28 39 40 41 42 43 44 45 47 86 114 117
1A81 5	1	13 15 16 19 41 43 48 51 60 64 68 70 71 74 84 86 87 88 109 114 117
1ABO 1	1	64 66 67 68 82 86 87 88 89 90 91 101 102 102 103 103 108 111 113 116
1ABO 2	1	64 65 65 66 67 84 87 88 89 90 91 93 100 101 102 103 108 111 113 116
1ABO 3	1	65 66 67 87 88 89 90 91 93 94 95 100 101 102 103 106 108 111 113 116
1ABO 4	1	64 65 66 67 69 87 88 89 90 91 93 100 101 102 103 106 108 111 113 116
1ABO 5	1	66 67 68 82 86 87 88 89 90 91 93 100 101 102 103 106 108 111 113 116
1BM2 1	1	55 56 60 61 62 83 84 85 86 87 95 97 99 110 118 125 127 133 150 152
1BM2 2	1	55 56 60 61 62 83 84 85 86 87 95 97 99 110 118 125 127 128 129 152
1BM2 3	1	55 56 58 60 61 62 64 67 69 73 83 84 85 86 87 129 132 133 150 152
1BM2 4	1	55 56 60 61 62 69 83 84 85 86 87 95 97 99 110 129 132 133 150 152
1BM2 5	1	58 60 60 61 61 62 64 67 69 73 75 83 84 85 86 129 132 133 150 152
1CKA 1	1	134 135 136 137 138 139 150 151 160 161 162 163 164 170 171 172 173 179 189 190
1CKA 2	1	134 135 136 137 139 150 151 153 160 161 162 163 164 170 171 172 173 179 189 190
1CKA 3	1	134 136 137 139 150 151 157 158 160 161 162 163 164 170 171 172 173 179 189 190
1CKA 4	1	136 137 139 150 151 153 158 159 160 161 162 163 164 170 171 172 173 179 189 190
1CKA 5	1	137 139 150 151 153 158 160 161 162 163 164 170 171 172 173 174 175 179 189 190
1G9O 1	1	9 10 11 13 14 15 31 34 38 54 57 58 60 68 90 91 92 94 95 96
1G9O 2	1	9 11 13 14 15 16 31 34 38 54 57 58 60 68 90 91 92 94 95 96
1G9O 3	1	9 11 13 14 15 31 34 38 54 55 57 58 60 68 90 91 92 94 95 96
1G9O 4	1	9 11 13 15 16 17 54 57 58 59 60 61 68 89 90 91 92 94 95 96
1G9O 5	1	10 11 13 15 16 17 54 57 58 60 61 68 89 90 90 91 92 94 95 96
1M61 1	1	34 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82
1M61 2	1	35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83
1M61 3	1	38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84 85 87
1M61 4	1	42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84 85 87 88 98
1M61 5	1	5 7 50 61 63 69 71 77 78 81 82 83 84 85 87 88 98 103 104 109
1O4C 1	1	32 33 34 35 43 45 63 65 71 73 79 80 81 82 83 84 85 86 87 89
1O4C 2	1	3 4 5 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 73 79
1O4C 3	1	1 3 4 5 6 7 8 9 11 17 31 32 33 35 43 45 65 81 82 83
1O4C 4	1	1 2 3 4 5 6 7 8 9 11 12 13 14 17 19 35 65 81 82 83
1O4C 5	1	1 3 4 5 6 7 8 9 11 12 17 31 32 33 34 35 65 81 82 83
1R6J 1	1	193 194 195 197 214 215 217 218 233 235 236 237 239 240 241 242 247 269 270 273
1R6J 2	1	193 194 197 198 199 217 233 235 236 237 238 239 240 241 242 247 268 270 272 273
1R6J 3	1	193 195 197 217 233 235 236 239 240 241 242 244 245 247 268 269 270 270 272 273
1R6J 4	1	193 195 197 217 233 235 236 237 239 241 242 244 245 247 268 269 270 272 273 273
1R6J 5	1	193 194 197 198 199 233 236 237 239 239 240 241 247 268 268 269 270 270 272 273
2BYG 1	1	186 187 188 189 190 191 192 215 216 219 244 246 270 271 273 274 278 280 281 282
2BYG 2	1	186 187 188 189 190 215 216 219 221 223 240 243 270 271 273 274 278 280 281 282
2BYG 3	1	186 187 188 189 190 215 216 219 221 223 240 243 244 270 271 273 278 280 281 282
2BYG 4	1	186 187 188 189 190 215 216 219 221 223 240 243 244 270 274 276 278 280 281 282
2BYG 5	1	187 189 190 191 192 215 216 219 221 223 240 243 244 270 274 276 278 280 281 282

Table 9 – Les tests avec vingt positions actives

Nom	S_{Vois}	positions actives
1A81 1	1	9 11 12 13 15 16 17 19 20 25 26 27 28 29 36 38 39 40 41 42 43 48 51 68 74 84 86 109 114 117
1A81 2	1	9 10 11 12 13 15 16 17 19 20 25 28 39 41 43 48 51 68 74 83 84 86 87 88 90 91 93 109 114 117
1A81 3	1	9 11 12 13 15 16 17 19 20 25 27 28 36 38 39 40 41 41 42 43 43 48 51 68 74 84 86 109 114 117
1A81 4	1	9 10 11 12 13 15 16 17 19 20 25 28 36 39 40 41 42 43 43 44 45 48 51 68 74 84 86 109 114 117
1A81 5	1	9 10 11 12 13 15 16 17 19 20 25 28 39 40 41 42 43 44 45 47 48 51 52 68 74 84 86 109 114 117
1ABO 1	1	64 65 66 67 68 70 71 72 75 78 79 80 81 82 83 86 87 88 89 90 91 93 100 101 102 103 108 111 113 116
1ABO 2	1	64 65 66 67 68 72 75 78 80 81 82 83 84 86 87 88 89 90 91 93 94 100 101 102 103 104 108 111 113 116
1ABO 3	1	64 66 67 68 70 71 72 78 82 86 87 88 89 90 91 93 94 95 96 99 100 101 102 103 104 105 108 111 113 116
1ABO 4	1	64 65 66 67 70 71 72 68 82 86 87 88 89 90 91 93 94 95 96 99 100 101 102 103 104 105 108 111 113 116
1ABO 5	1	65 66 67 70 71 72 75 78 80 82 86 87 88 89 90 91 93 94 95 96 99 100 101 102 103 108 111 113 116
1BM2 1	1	55 56 58 60 61 62 83 84 85 86 87 95 97 99 110 118 119 120 121 122 125 127 128 129 130 131 132 133 150 152
1BM2 2	1	56 58 60 61 62 83 84 85 86 87 95 97 99 110 118 119 120 121 122 123 125 127 128 129 130 131 132 133 150 152
1BM2 3	1	58 60 61 62 83 84 85 86 87 95 97 99 108 109 110 118 120 121 122 123 125 127 128 129 132 133 134 135 150 152
1BM2 4	1	55 56 58 60 61 62 83 84 85 86 87 95 97 99 108 109 110 118 120 121 125 127 128 129 130 131 132 133 150 152
1BM2 5	1	56 58 60 61 62 67 83 84 85 86 87 95 97 99 110 111 112 113 115 118 125 127 128 129 130 131 132 133 150 152
1CKA 1	1	134 135 136 137 139 140 141 142 143 144 146 147 148 149 150 151 158 159 160 161 162 163 164 170 171 172 173 179 189 190
1CKA 2	1	134 135 136 137 139 143 144 146 147 148 149 150 151 158 159 160 161 162 163 164 170 171 172 173 179 186 187 188 189 190
1CKA 3	1	135 136 137 139 144 146 147 148 149 150 151 157 158 159 160 161 162 163 163 164 170 171 172 173 179 186 187 188 189 190
1CKA 4	1	136 137 139 140 141 142 143 144 146 147 148 151 158 159 160 161 162 163 164 170 171 172 173 179 184 186 187 188 189 190
1CKA 5	1	134 136 137 139 140 141 142 143 144 146 147 148 151 158 159 160 161 162 163 164 170 171 172 173 179 182 187 188 189 190
1G9O 1	1	9 10 11 13 15 24 31 34 38 40 41 42 43 46 48 49 50 51 54 57 58 60 68 89 90 91 92 94 95 96
1G9O 2	1	9 11 13 15 31 32 34 38 40 41 42 43 46 48 49 50 51 54 57 58 60 68 89 90 90 91 92 94 95 96
1G9O 3	1	9 10 11 13 15 31 32 34 38 40 41 42 43 46 48 49 50 51 54 57 58 60 68 89 90 91 92 94 95 96
1G9O 4	1	10 11 13 14 15 31 32 34 38 40 41 42 43 46 48 49 50 51 54 57 58 60 61 68 89 90 91 92 94 95 96
1G9O 5	1	10 11 13 14 15 31 32 40 41 42 43 46 48 49 50 51 54 57 58 60 61 62 68 87 89 90 91 92 94 95 96
1M61 1	1	12 14 15 20 34 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84 85 87 88 98
1M61 2	1	6 7 8 10 11 12 14 15 20 21 34 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82
1M61 3	1	5 7 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84 85 87 88 98 103 104 109
1M61 4	1	7 8 10 11 12 14 15 20 34 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84
1M61 5	1	8 10 11 12 14 15 20 34 35 36 37 38 39 42 46 47 48 49 50 61 63 69 71 77 78 81 82 83 84 85
1O4C 1	1	1 2 3 4 5 6 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 81 82 83 90 91 92 93 94 96
1O4C 2	1	1 3 4 5 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 73 79 80 81 82 83 90 91 92 93 96
1O4C 3	1	1 3 4 5 6 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 73 79 80 81 82 83 90 91 92 93 96
1O4C 4	1	1 3 4 5 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 73 79 80 81 82 83 84 91 92 93 96
1O4C 5	1	1 3 4 5 6 7 8 9 11 17 31 32 33 34 35 43 45 63 65 71 73 79 80 81 82 83 84 92 93 96
1R6J 1	1	193 194 195 197 198 199 217 218 219 220 221 222 223 224 225 226 227 228 229 233 235 236 237 239 247 268 269 270 272 273
1R6J 2	1	193 194 195 197 198 199 217 220 221 222 223 224 225 226 227 228 229 233 235 235 236 237 239 240 247 268 269 270 272 273
1R6J 3	1	193 194 195 197 198 199 208 217 220 221 222 223 224 225 226 227 228 229 230 233 235 236 237 239 247 268 269 270 272 273
1R6J 4	1	193 194 195 197 198 199 217 220 221 222 223 224 225 226 227 228 229 233 235 236 237 239 240 247 249 268 269 270 272 273
1R6J 5	1	194 195 197 198 199 217 220 221 222 223 224 225 226 227 228 229 233 235 236 237 239 240 247 249 250 268 269 270 272 273
2BYG 1	1	186 187 188 189 190 191 192 215 216 219 221 223 240 243 244 246 250 251 252 253 254 255 256 257 259 260 278 280 281 282
2BYG 2	1	186 187 188 189 190 191 192 198 215 216 219 221 223 240 243 244 251 252 253 254 255 256 257 259 260 278 278 280 281 282
2BYG 3	1	186 187 188 189 190 190 191 192 215 216 219 221 223 240 243 244 251 252 253 254 255 256 257 259 260 278 246 280 281 282
2BYG 4	1	186 187 188 189 190 191 192 215 216 219 221 223 240 243 244 245 246 251 252 253 254 255 256 257 259 260 278 278 281 282
2BYG 5	1	186 187 188 189 190 191 192 215 216 219 221 223 240 243 244 251 252 253 254 255 256 257 259 260 278 246 278 280 281 282

Table 10 – Les tests avec trente positions actives

Résum

Titre de la thèse

XXX

Mots-clés : motclé1, motclé2, motclé3

Abstract

Thesis title

XXX

Keywords: keyword1, keyword2, keyword3