



# Computational Protein Design: un outil pour l'ingénierie des protéines et la biologie synthétique

**David Mignon**

sous la direction de Thomas Simonson  
Laboratoire de Biochimie, **École Polytechnique**

# Computational Protein Design (CPD)

Concevoir ou modifier des protéines par informatique pour leur conférer de nouvelles propriétés

Application:

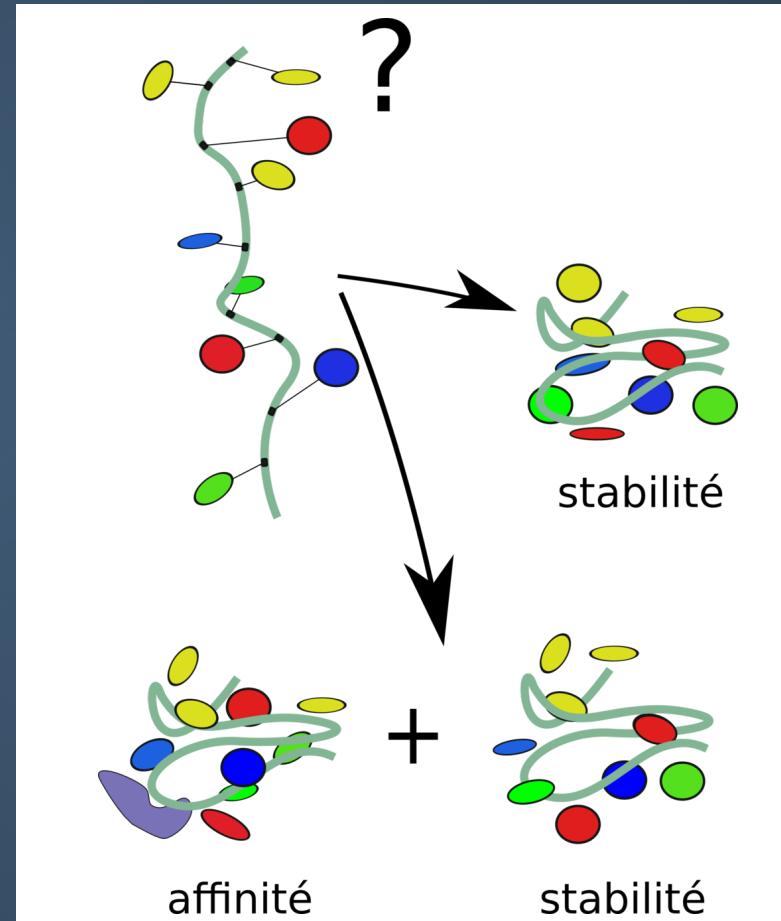
- protéines entièrement redessinées
- enzymes, substrat, complexes

Principaux éléments:

- un espace de conformations de la protéine
- une fonction d'énergie
- un algorithme d'exploration de l'espace de séquences-conformations

Principaux programmes:

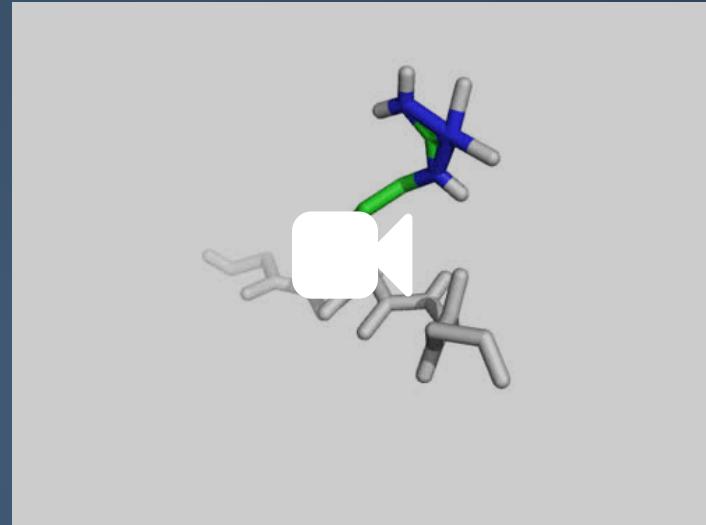
- ORBIT (Mayo, 1996)
- OSPREY(Donald)
- Toulbar2 (Schiex)
- Proteus (Simonson)
- Rosetta (Baker)



# Le CPD avec Proteus

## 1. L'espace de conformations:

- Le squelette d'une protéine native
- Un backbone fixé
- positionnement des chaînes latérales discrétisées (rotamères)
- Utilisation d'une bibliothèque de rotamères: Tuffery 95



## 2. L'état déplié: l'énergie de référence

Pour une séquence  $S$  de type  $t_i$

$$E^u(S) = \sum_i^N E_{t_i}^u$$

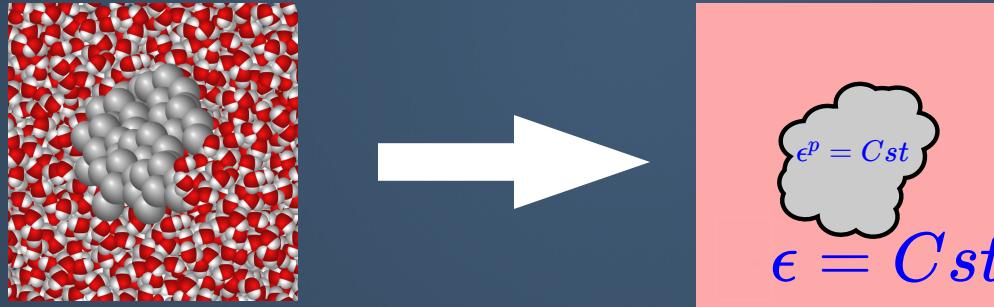
Une position mutable

# La fonction d'énergie

- L'énergie interne à la protéine:

Utilisation de la mécanique moléculaire avec le champ de force AMBER (ff99SB)

- Les modèles de solvant implicites



1. L'effet hydrophobe: modèle "Surface Area" (SA)
2. Traitement des interactions électrostatiques:  
Coulomb (CA) ou Generalised Born (GB)

# Objectifs et Résultats

## L' Exploration dans Proteus

- Le Monte Carlo (mono et multi-marcheurs)
- Comparaison d'algorithmes

## Optimisation et benchmarks

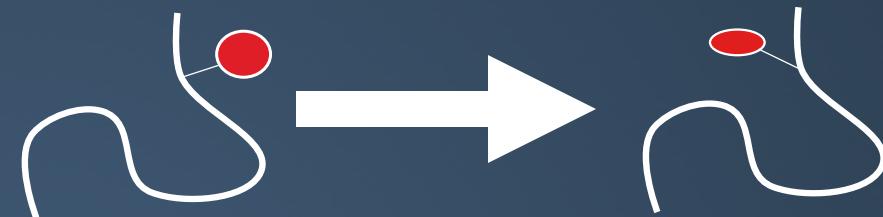
- Les énergies de référence
- Le GB/FDB
- Protéines PDZ et protocoles
- Évaluation et comparaison de nos séquences
- Méthode de croissance du noyau hydrophobe

# Principe de l'exploration

Déplacement dans l'espace des conformations:

modification du rotamère à une position i sur la protéine repliée

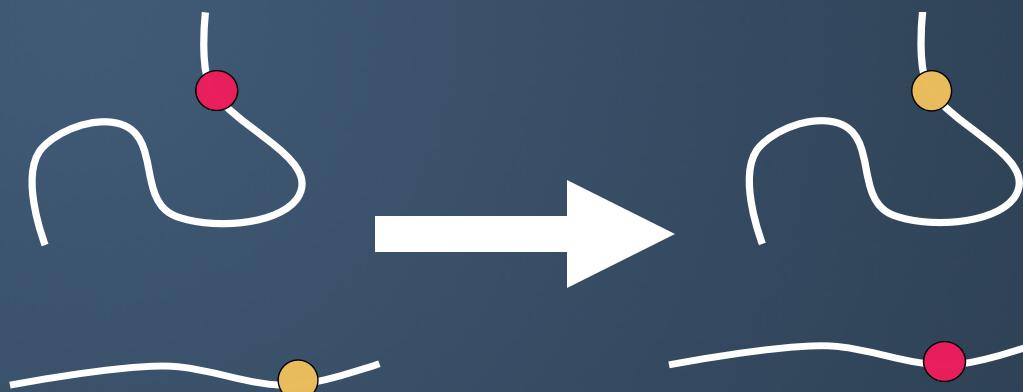
$$\Delta E = E(.., rot_i^{new}, ..) - E(.., rot_i^{old}, ..)$$



Déplacement dans l'espace des séquences:

modification du type de chaîne latérale à une position i sur la protéine repliée.

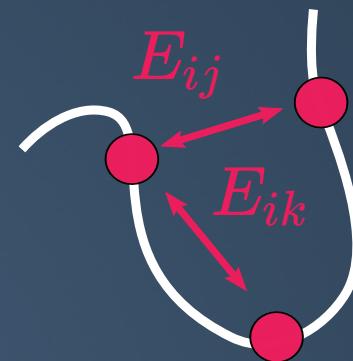
En même temps, une mutation inverse sur la protéine dépliée, en i



$$\Delta E = \Delta E_f - \Delta E_{uf}$$

# Décomposition par paires de la fonction d'énergie

$$E(C) = \sum_i E_i + \sum_{i \neq j} E_{ij}$$



- rendre possible l'utilisation de plusieurs algorithmes
- accélérer l'étape d'exploration par pré-calcul des interactions -> stockage dans une matrice.
- nécessite des approximations supplémentaires:
  1. Pour la décomposition du terme surfacique
  2. Pour la décomposition de l'environnement GB

*Première méthode:* "Native Environment Approximation" (NEA)

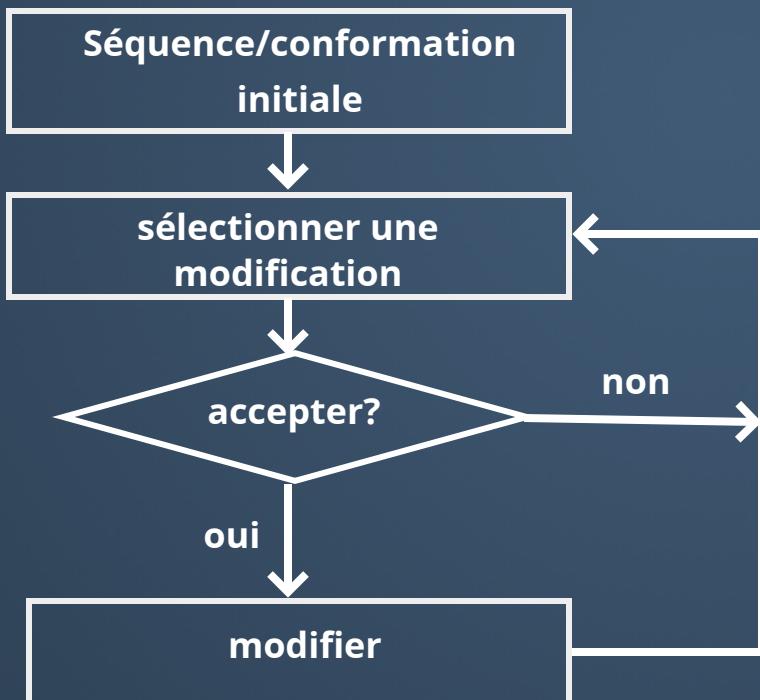
Les rayons de solvatation d'une chaîne latérale sont calculés en fixant tout le reste du système dans sa séquence et sa conformation native.

# Le Monte Carlo

algorithme Metropolis-Hastings

L'objectif est de générer une collection d'états échantillonnés selon la distribution de Boltzmann:

$$\mathcal{P}(x) = \frac{1}{Z} e^{-\frac{E}{RT}}$$



Un pas depuis l'état  $x$  est contrôlé par 2 probabilités conditionnelles:

- $\sigma(\cdot|x)$  pour le choix de la modification
- $\alpha(\cdot|x)$  pour l'accepter

# Le Monte Carlo

L'algorithme définit une chaîne de Markov.

Si la balance détaillée est respectée

$$P(x \rightarrow y) = P(y \rightarrow x)$$

alors la chaîne possède une distribution stationnaire.

Si la chaîne est ergodique, elle converge vers cette distribution.

Metropolis propose pour  $\sigma$  symétrique:

$$\alpha(y|x) = \min\left\{1, \frac{\mathcal{P}(y)}{\mathcal{P}(x)}\right\} = \min\left\{1, e^{(-\frac{\Delta E}{RT})}\right\}$$

# Le Monte-Carlo

## Algorithme Metropolis-Hastings

Une séquence-conformation  $S$  est choisie aléatoirement

Pour chaque pas de la trajectoire

Une modification possible est sélectionnée à partir d'une distribution conditionnelle:

$S' \sim \sigma(S'|S)$  choix de positions, types, rotamères

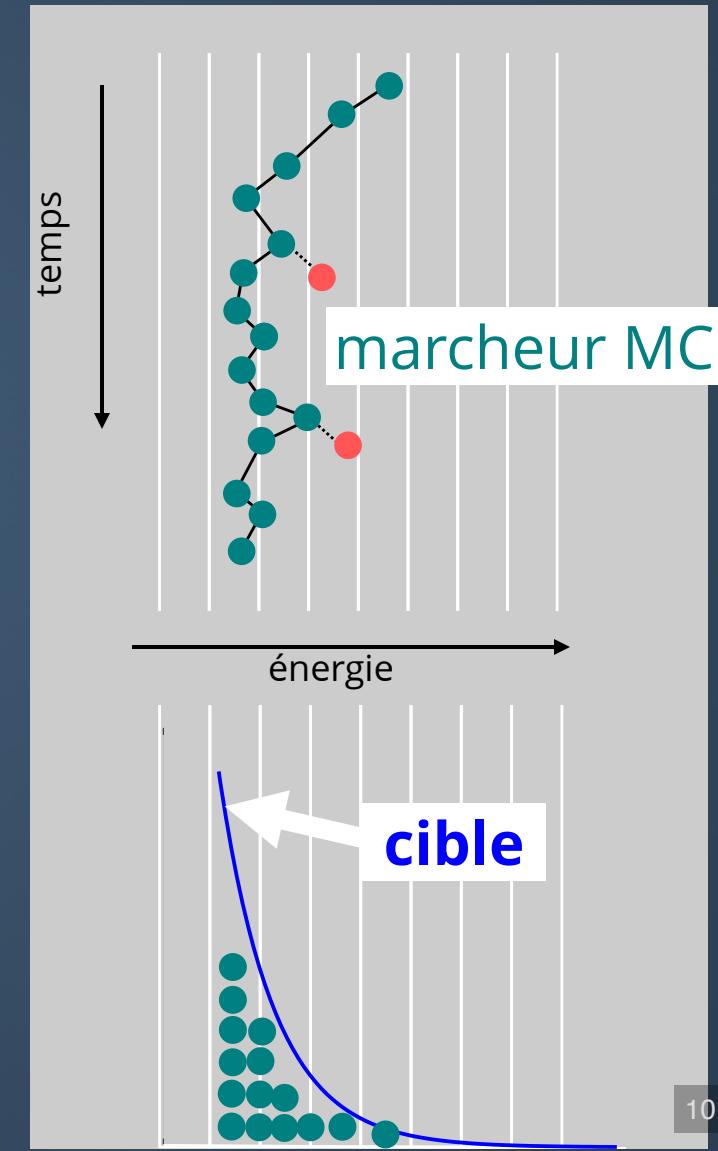
La modification est acceptée selon la probabilité:

$$\alpha(S'|S) = \min\left\{1, e^{\left(-\frac{\Delta E}{RT}\right)} \frac{\sigma(S|S')}{\sigma(S'|S)}\right\}$$

Fin de Pour

indépendant de la  
fonction de partition

généralisation  
d'Hastings



# Replica Exchange Monte Carlo

accélérer la convergence en visitant plusieurs zones énergétiques simultanément

Lancement en parallèle de  $n$  marcheurs Monte Carlo aux températures ordonnées  $(t_1, \dots, t_n)$

Périodiquement ( $m \propto M$ ) un couple de marcheurs aux températures  $(t_i, t_{i+1})$  est sélectionné

Les températures entre les deux marcheurs sont échangées selon la probabilité:

$$\beta(X_{m+1}|X_m) = \min\{1, \exp(-(\frac{1}{kt_i} - \frac{1}{kt_{i+1}})(E_{t_i} - E_{t_{i+1}}))\}$$

- L'échange de températures est une mouvement supplémentaire dans l'espace généralisé des  $n$  répliques.
  - $\beta$  provient de l'expression de la balance détaillée pour ce mouvement
- Donc les propriétés Monte Carlo de la trajectoire sont conservées.

# Le "Multistart Steepest Descent" (Wernisch,Wodak)

Une heuristique spécifique à l'espace d'états

Pour chaque cycle heuristique

une séquence-conformation **S** est choisie aléatoirement

Tant que l'énergie de **S** est améliorée

Pour **i** allant de la première position de **S** jusqu'à la dernière

**S** est fixée sauf à la position **i**

le meilleur rotamère possible en **i** est déterminé

Ce rotamère est utilisé pour fixer **S** en **i**

Fin de Pour

Fin de Tant que

**S** est sauvegardée

Fin du cycle heuristique



# Le "Multistart Steepest Descent" (Wernisch,Wodak)

Une heuristique spécifique à l'espace d'états

Pour chaque cycle heuristique

une séquence-conformation **S** est choisie aléatoirement

Tant que l'énergie de **S** est améliorée

Pour **i** allant de la première position de **S** jusqu'à la dernière

**S** est fixée sauf à la position **i**

le meilleur rotamère possible en **i** est déterminé

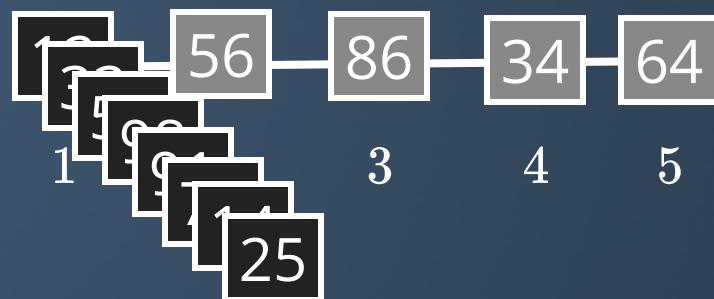
Ce rotamère est utilisé pour fixer **S** en **i**

Fin de Pour

Fin de Tant que

**S** est sauvegardée

Fin du cycle heuristique



# Le "Multistart Steepest Descent" (Wernisch,Wodak)

Une heuristique spécifique à l'espace d'états

Pour chaque cycle heuristique

une séquence-conformation **S** est choisie aléatoirement

Tant que l'énergie de **S** est améliorée

Pour **i** allant de la première position de **S** jusqu'à la dernière

**S** est fixée sauf à la position **i**

le meilleur rotamère possible en **i** est déterminé

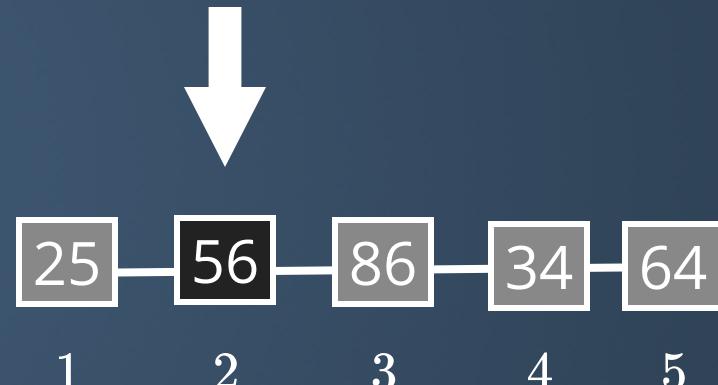
Ce rotamère est utilisé pour fixer **S** en **i**

Fin de Pour

Fin de Tant que

**S** est sauvegardée

Fin du cycle heuristique



# Le "Multistart Steepest Descent" (Wernisch,Wodak)

Une heuristique spécifique à l'espace d'états

Pour chaque cycle heuristique

une séquence-conformation **S** est choisie aléatoirement

Tant que l'énergie de **S** est améliorée

Pour **i** allant de la première position de **S** jusqu'à la dernière

**S** est fixée sauf à la position **i**

le meilleur rotamère possible en **i** est déterminé

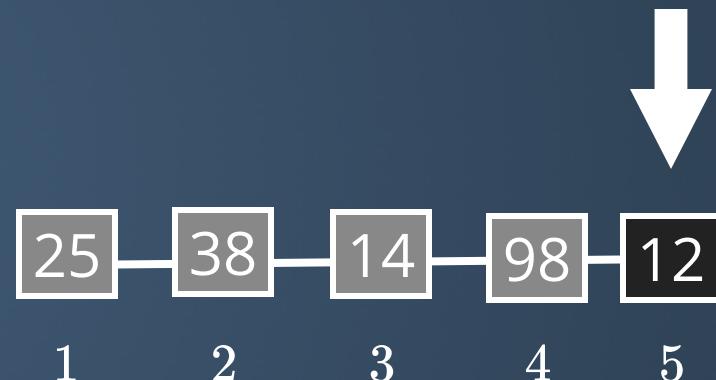
Ce rotamère est utilisé pour fixer **S** en **i**

Fin de Pour

Fin de Tant que

**S** est sauvegardée

Fin du cycle heuristique



# Une méthode exacte de recherche du minimum global/GMEC (Toulbar2)

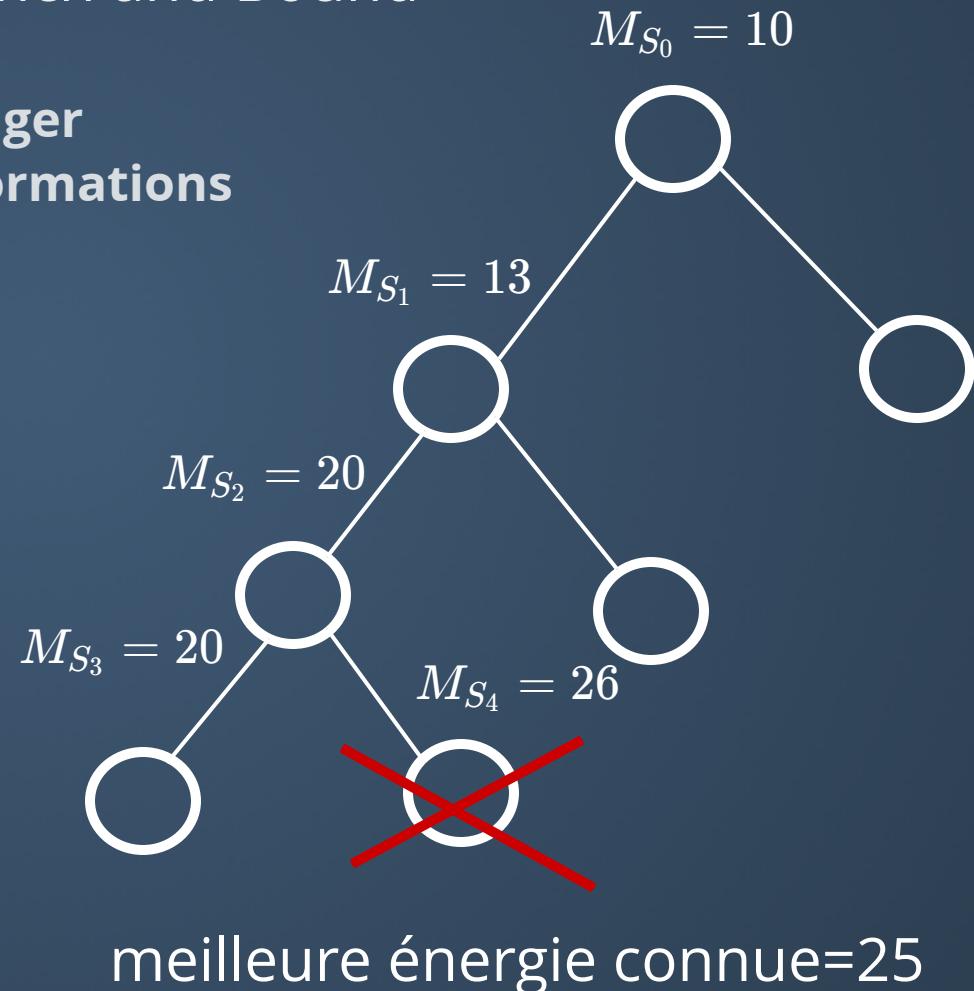
L'algorithme "Depth-First Branch and Bound"

- **Le principe de séparation:** partager l'ensemble des séquences-conformations en sous-ensemble fils.

→ construction d'un arbre

- **La recherche dans l'arbre:** descendre dans les branches autant que possible, sinon remonter d'un sommet.

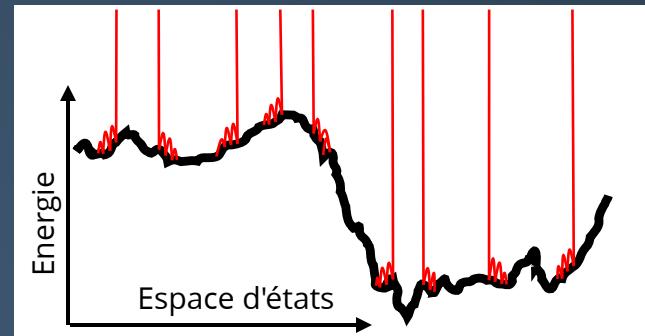
- **Calculer les minorants  $M_S$  des énergies des sommets.**
- **Si une énergie connue est inférieure à un minorant  $M_S$  on peut élaguer l'arbre en  $S$**



# Différentes approches

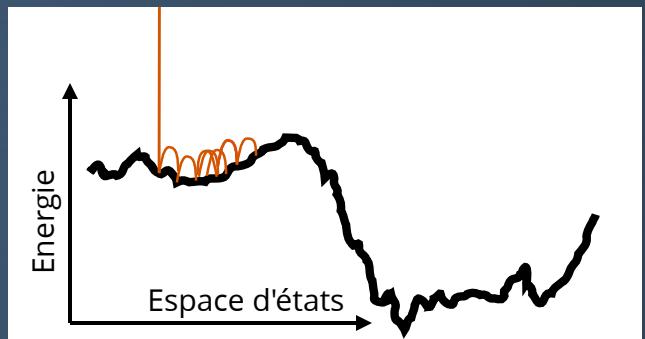
## MSD

- Pour un cycle, l'exploration est limitée.
- Mais un cycle est rapide.



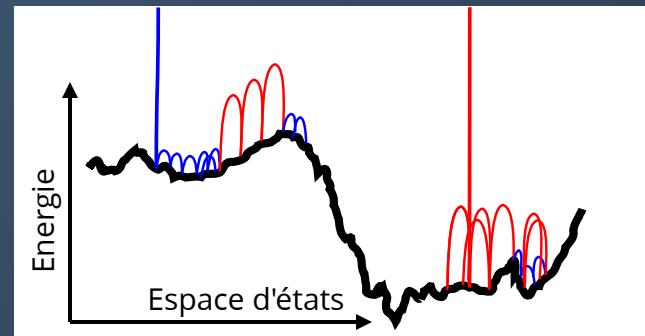
## Monte Carlo

- Le marcheur peut tout visiter.
- Mais la convergence est lente.



## REMC

- Il conserve les propriétés physique de la trajectoire.
- La convergence est accélérée.



# Objectifs et résultats

## L' Exploration dans Proteus

- le Monte Carlo (mono et multi-marcheurs)
- **Comparaison d'algorithmes**

## Optimisation et benchmarks

- Les énergies de référence
- Le GB/FDB
- Protéines PDZ et protocoles
- Évaluation et comparaison de nos séquences
- Méthode de croissance du noyau hydrophobe

# Comparaison des algorithmes

L'ensemble de tests

## Systèmes

Protéine	nb résidus	famille
1A81	108	SH2
1BM2	98	SH2
1M61	109	SH2
1O4C	104	SH2
1ABO	58	SH3
1CKA	57	SH3
1G9O	91	PDZ
1R6J	82	PDZ
2BYG	97	PDZ

## Modèle

- Amber (ff99SB)
- CASA,  $\epsilon = 23$
- toutes les positions mutables, sauf GLY et PRO
- énergies de références optimisées sur les 3 familles

# Comparaison des algorithmes méthodes

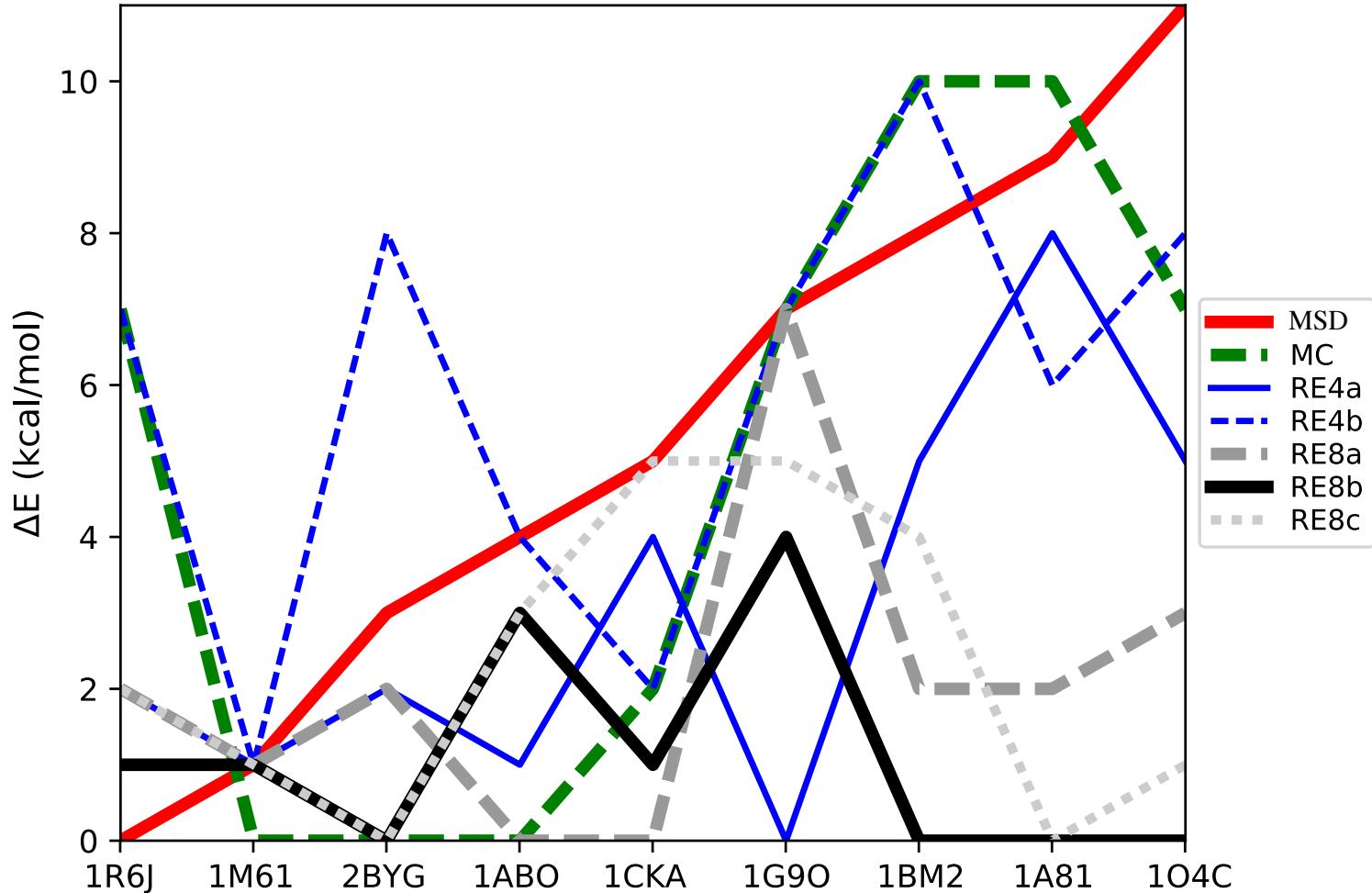
temps d'exécution limité à 24h

- MSD: 110 000 cycles
- Monte carlo: 6 milliards de pas
- REMC: 6 milliards de pas cumulés sur les marcheurs en parallèle (openMP)
- Un déplacement MC/REMC est constitué d'un changement de rotamère, d'une mutation ou d'un couple.

## Paramétrages

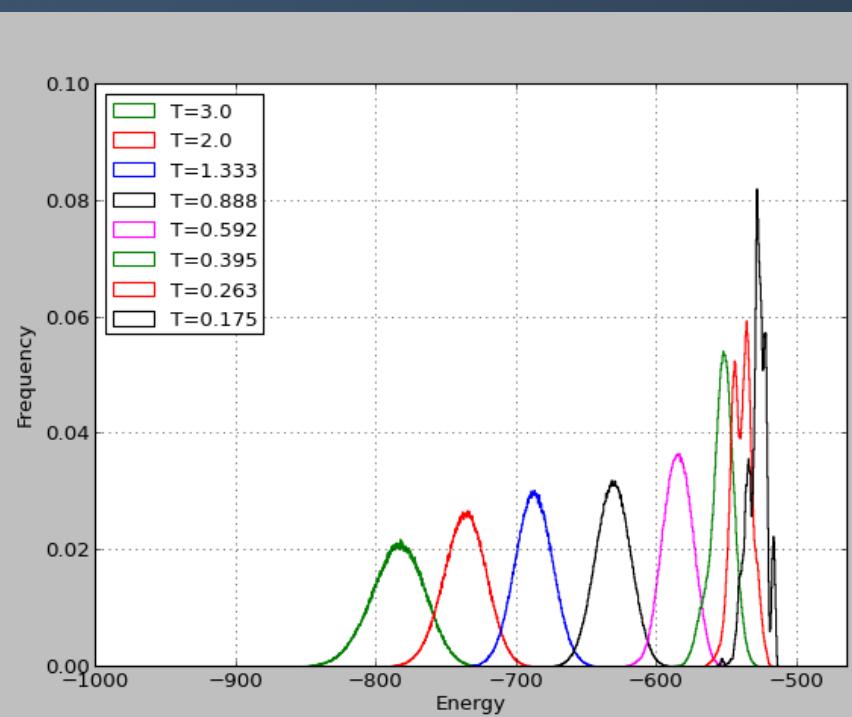
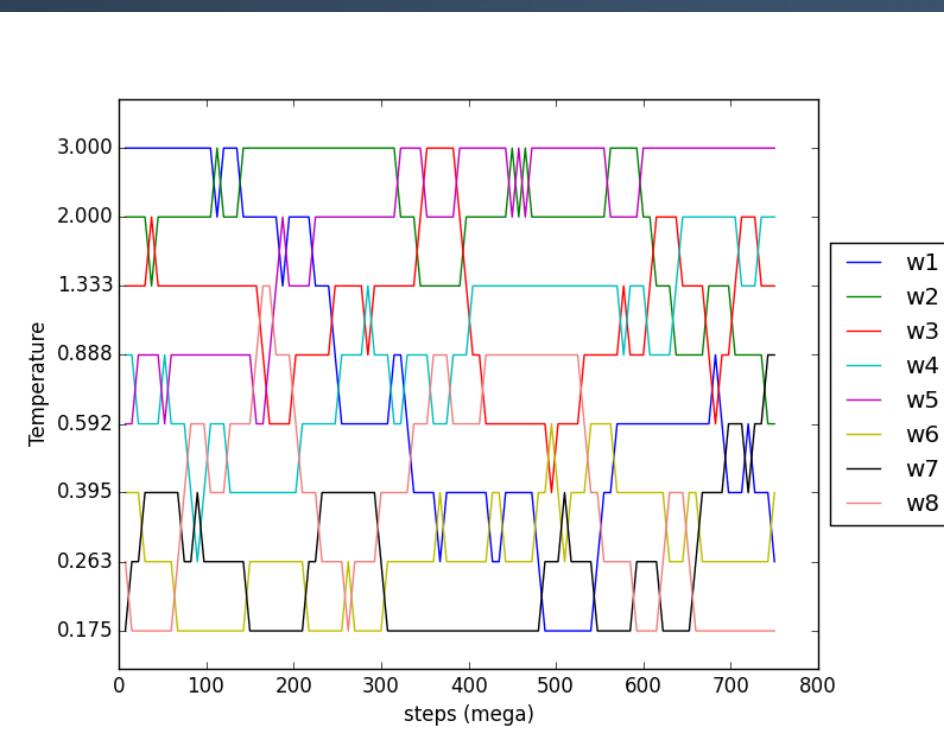
Algo	nombre de marcheurs	températures	mutation / pas	change de rotamères/ pas	freq swap
MC	1	0,2	1,1	0,1	-
REMC	4	0,125...1	0,1	1,1	0,005
REMC	4	0,25...2	0,1	1,1	0,005
REMC	8	0,175...3	1,1	0,1	0,01
REMC	8	0,175...3	0,1	1,1	0,01
REMC	8	0,175...3	0,1	1,1	0,001

# Comparaison des meilleures énergies



# REMC - Distribution d'énergie

1A81 tout actif

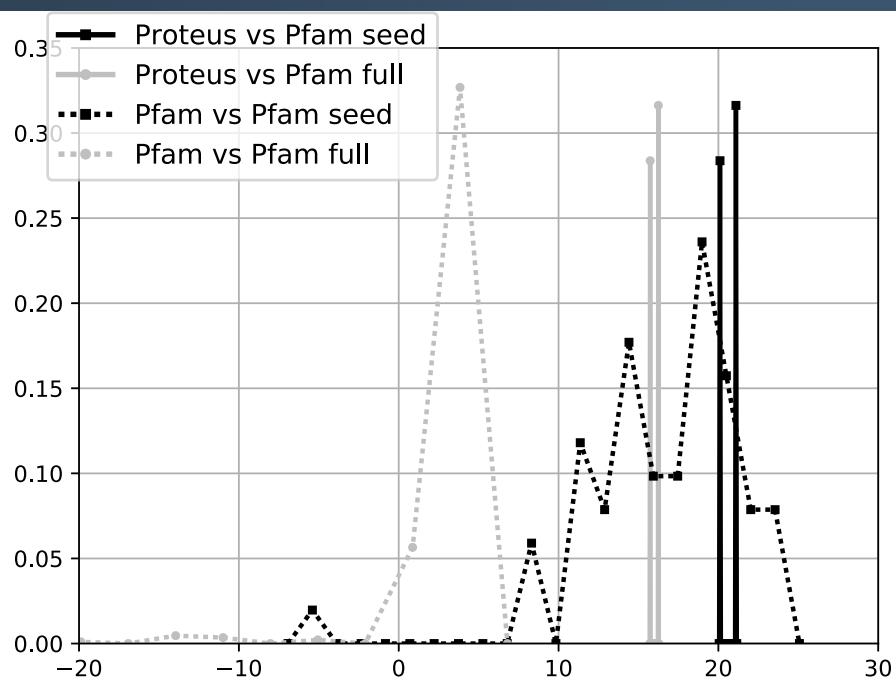


# Caractérisation des séquences

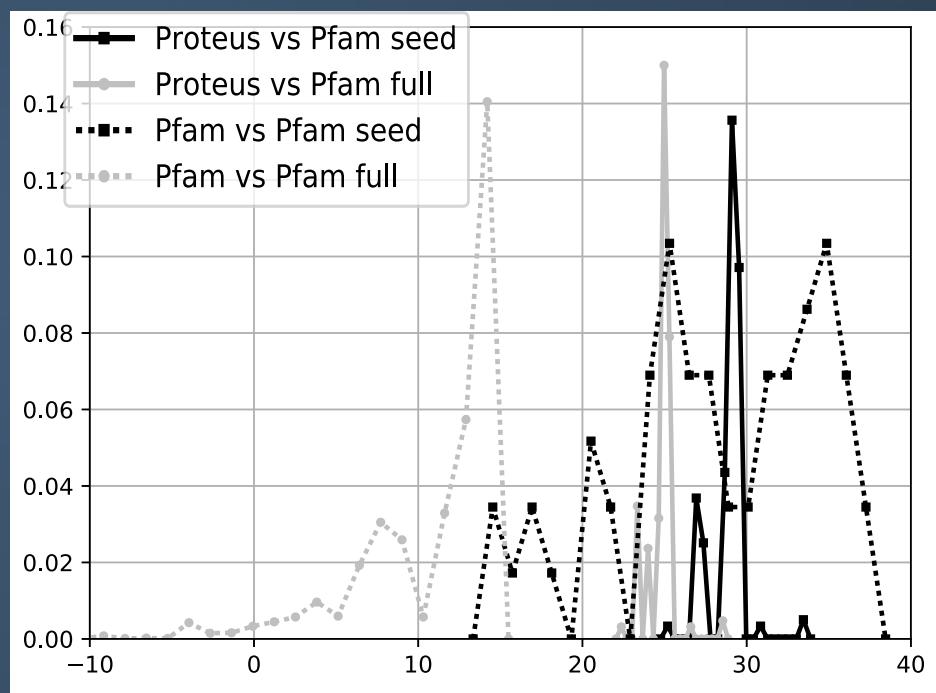
- Calcul de similarité aux alignements "seed" et "full" de Pfam (Protein families database) de la famille correspondante, aux positions du cœur (BLOSUM64)
- Soumission à Superfamily reconnaissance de structure 3D à partir d'une bibliothèque de HMM obtenue grâce à la classification SCOP
- Taux d'identité à la séquence native (backbone)

# Scores de similarité sur les positions du cœur

1ABO



1BM2



# Résultats Superfamily et identité

## sur les 10000 séquences-conformations de meilleures énergies

Protéine	nb de séquences	% identité à la native	taille du "match"	E-value Super-famille	% succès Super-famille	E-value famille	taux succès famille
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%	4.6e-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%

# Recherche du GMEC

l'Espace est réduit progressivement en fixant une partie des résidus avec leur type natif:

- Tests à 30, 20, 10, 5 et 1 positions actives
- Dans chaque cas 5 sélections en privilégiant les ensembles en interaction pour chacune des 9 protéines.

Toulbar2: Temps d'exécution max 24h, en cas d'échec relance avec une seconde configuration

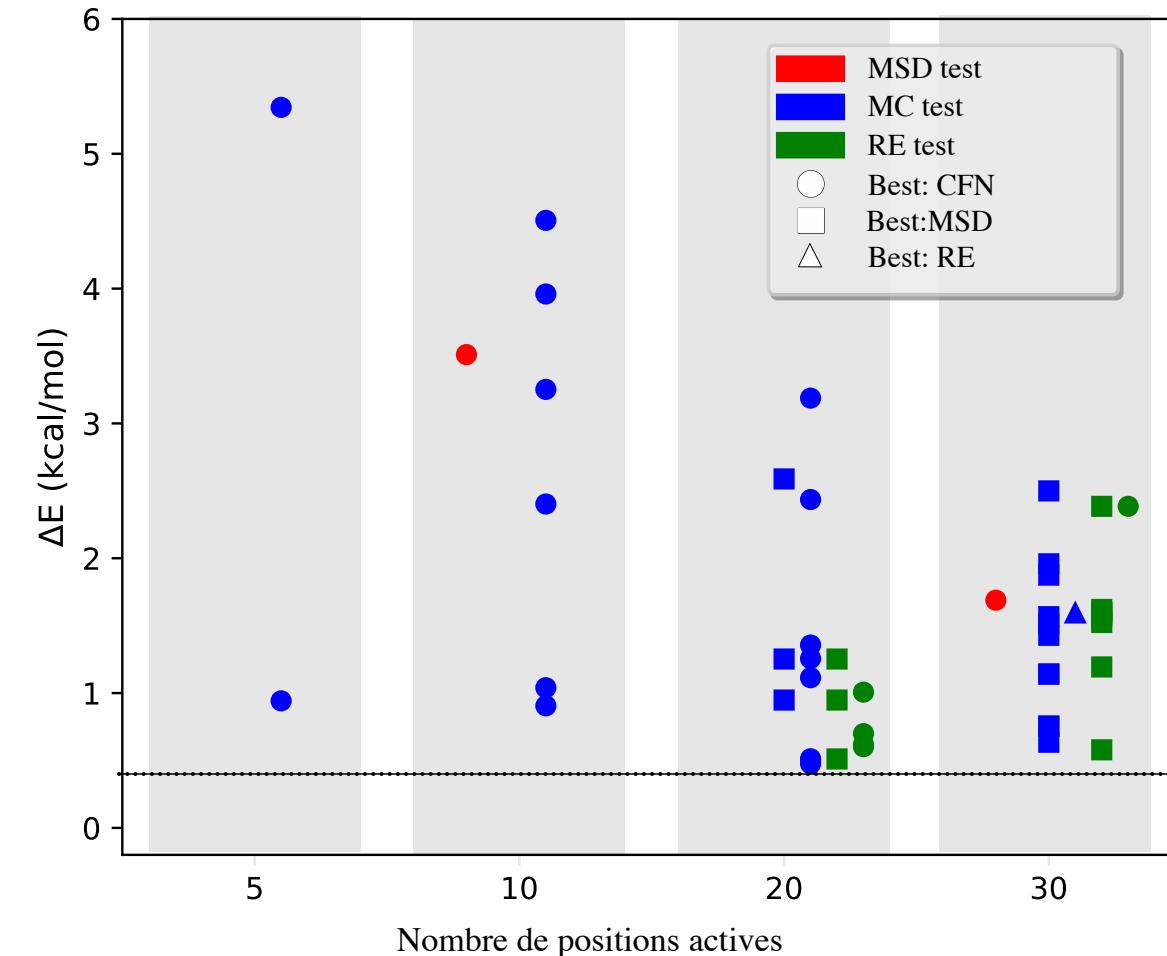
Nos algorithmes: MSD et MC, le meilleur REMC si utile.

# Résultats

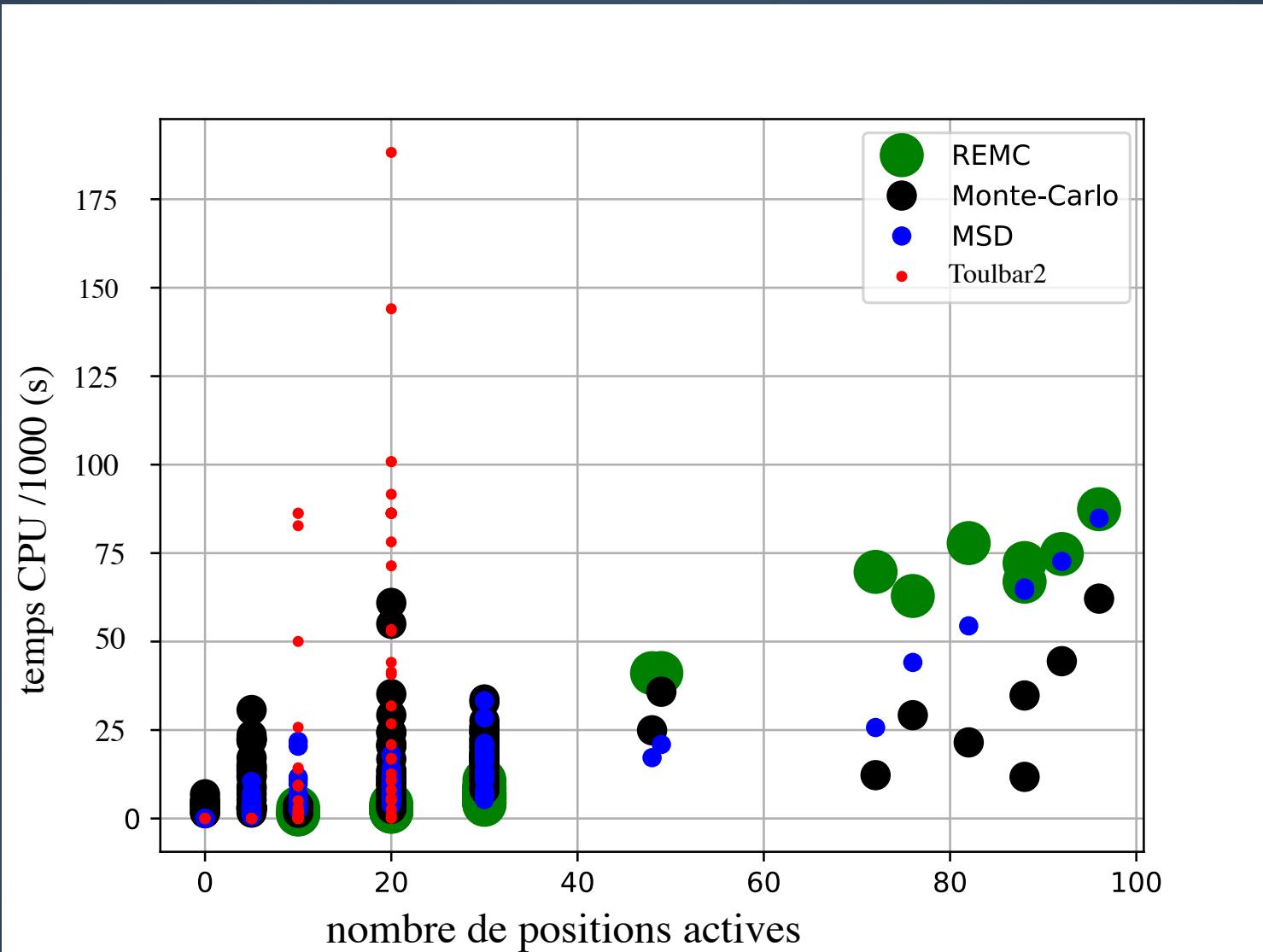
- 10 positions actives ou moins: le GMEC est établi pour tous les tests sauf 1.  
20 positions actives: 27/45 GMEC, 30 positions actives: 1/45
- Le MSD trouve presque tous les GMEC connus et domine sur quasiment tous les autres tests.
- Le MC/REMC est presque toujours à moins de 2 kcal/mol du meilleur résultat.

# Résultats

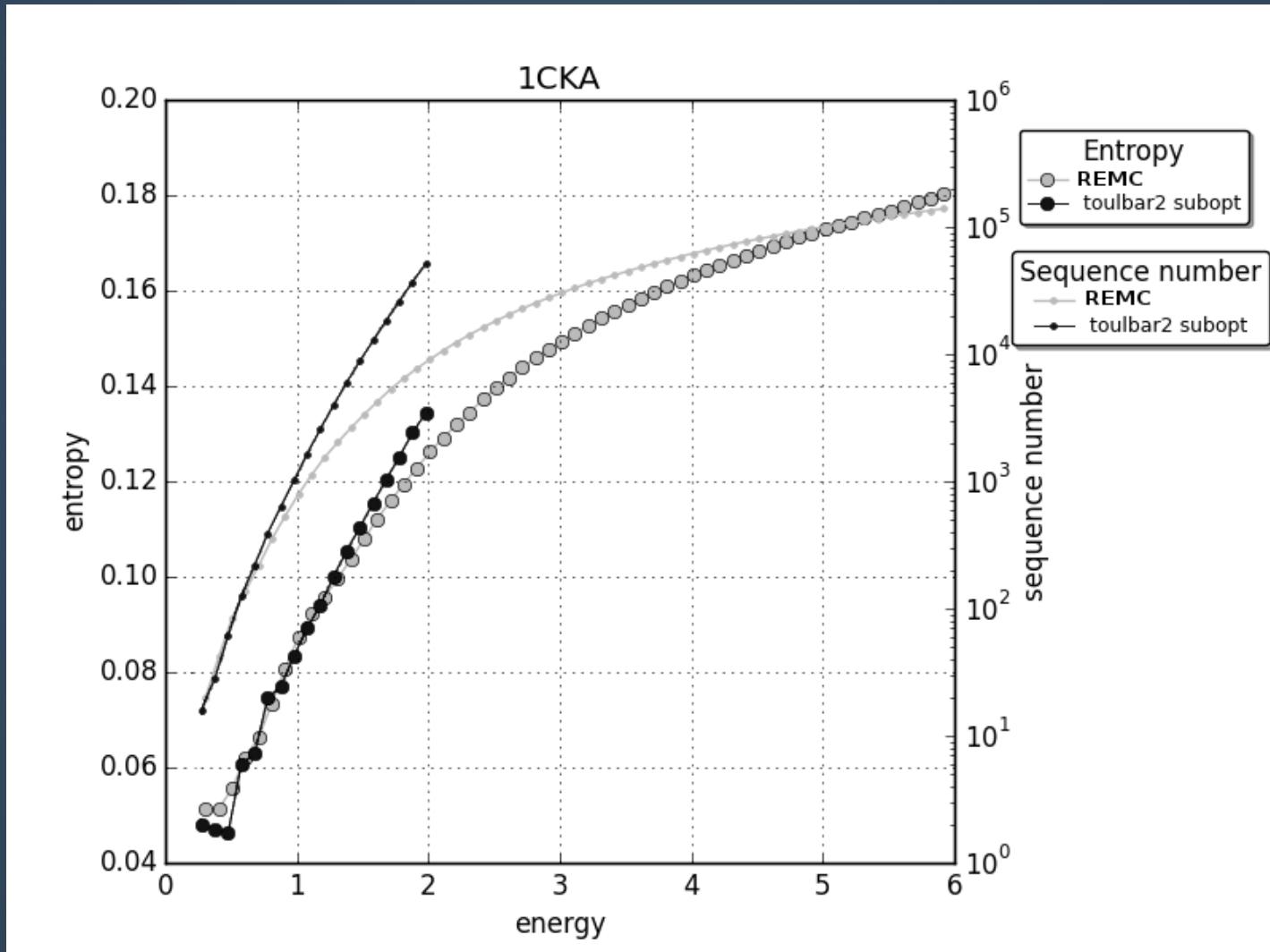
## Différences avec la meilleure énergie



# Temps CPU

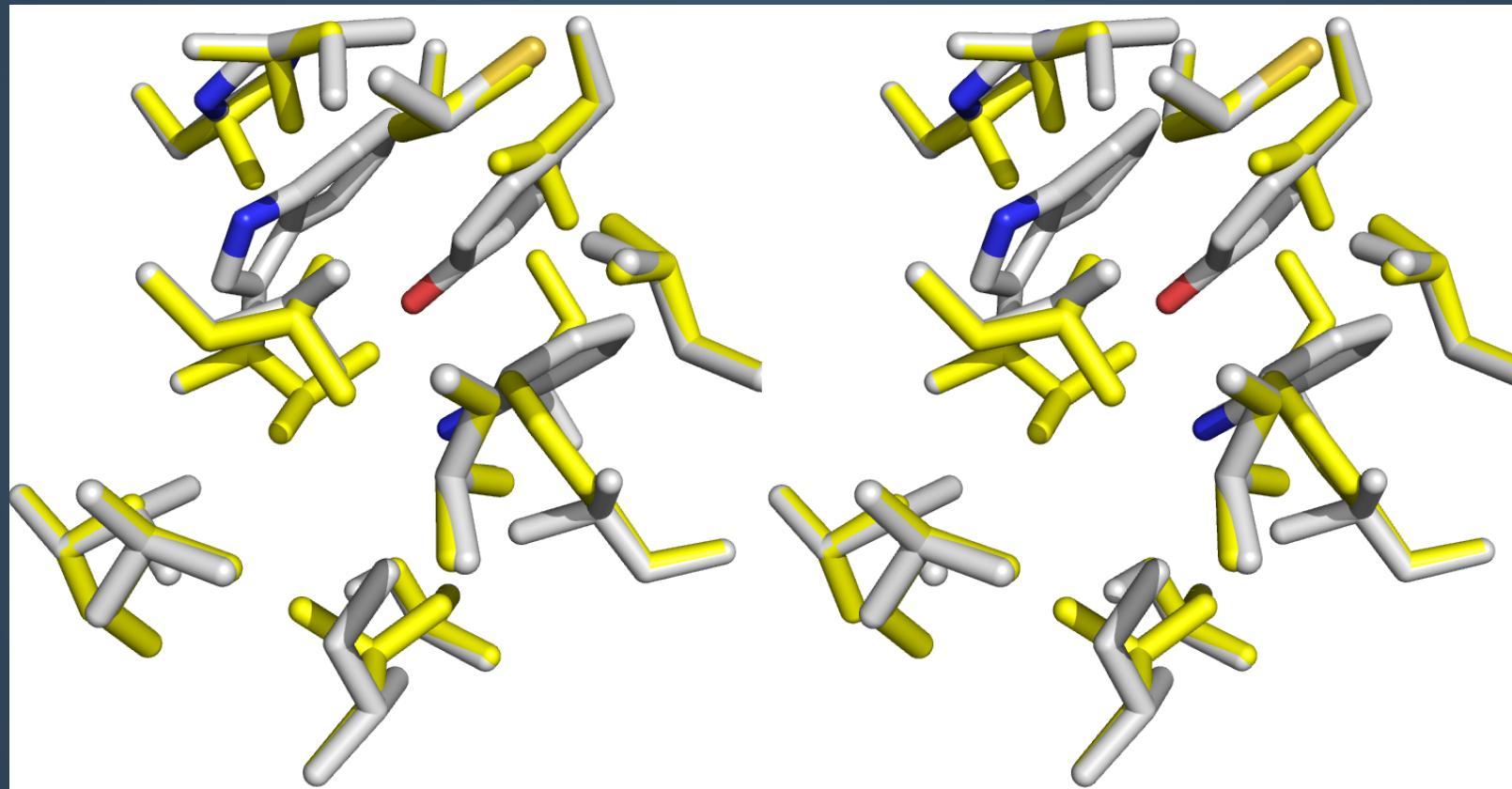


# Densité d'états au voisinage du GMEC



# Exemple de structure obtenue 2BYG 10 positions actives

expérimentale (jaune) - Proteus/MSD(blanc)



# Objectifs et Résultats

## L' Exploration dans Proteus

- Le Monte Carlo (mono et multi-marcheurs)
- Comparaison d'algorithmes

## Optimisation et benchmarks

### • **Les énergies de référence**

- Le GB/FDB
- Protéines PDZ et protocoles
- Évaluation et comparaison de nos séquences
- Méthode de croissance du noyau hydrophobe

# Optimisations des énergies de références

L'énergie de l'état déplié d'une séquence  $S$  est de la forme:

$$E_s^u = \sum_{i \in s} E_{t_i}^r$$



Ce sont des paramètres ajustables

L'objectif est de reproduire des fréquences d'acides aminés cibles.

Les homologues naturels donnent les fréquences cibles.

3 algorithmes sont utilisés:

- Une méthode d'ajustement logarithmique:

$$E_t^r(n+1) = E_t^r(n) + c \ln(freq_t^{MC}(n)/freq_t^{naturelle})$$

- La méthode du gradient
- Une méthode du gradient à pas variable

# Maximiser la vraisemblance des énergies de références par la méthode du gradient

Soit  $\mathcal{S}$  un ensemble de séquences naturelles,  $P(\mathcal{S})$  sa probabilité de Boltzmann est une fonction des  $E_t^r$

Nous cherchons les  $E_t^r$  qui maximisent  $P(\mathcal{S})$ , elles réalisent notre objectif.

La méthode consiste à avancer dans la direction du gradient de  $\ln(P(\mathcal{S}))$

On a  $\frac{1}{N} \frac{\partial}{\partial E_t^r} \ln P(\mathcal{S}) = freq_t^{\mathcal{S}} - \langle n(t) \rangle$

L'algorithme itératif:

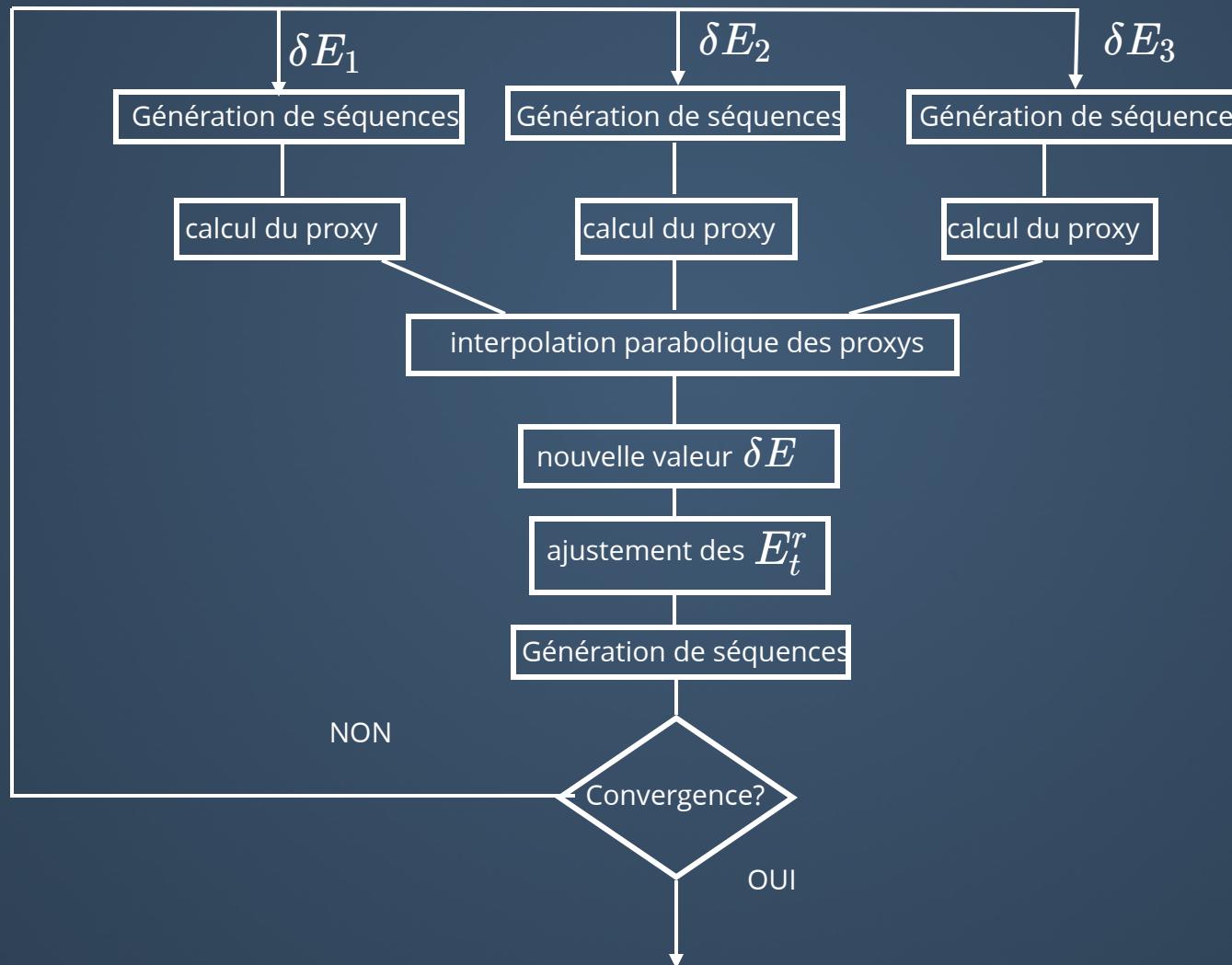
 moyenne de Boltzmann

$$E_t^r(n+1) = E_t^r(n) + \delta E \times (freq_t^{\mathcal{S}} - freq_t^{MC}(n))$$

# La méthode du gradient à $\delta E$ variable

Une fonction proxy  $C$  comme outil de mesure de l'état de l'optimisation

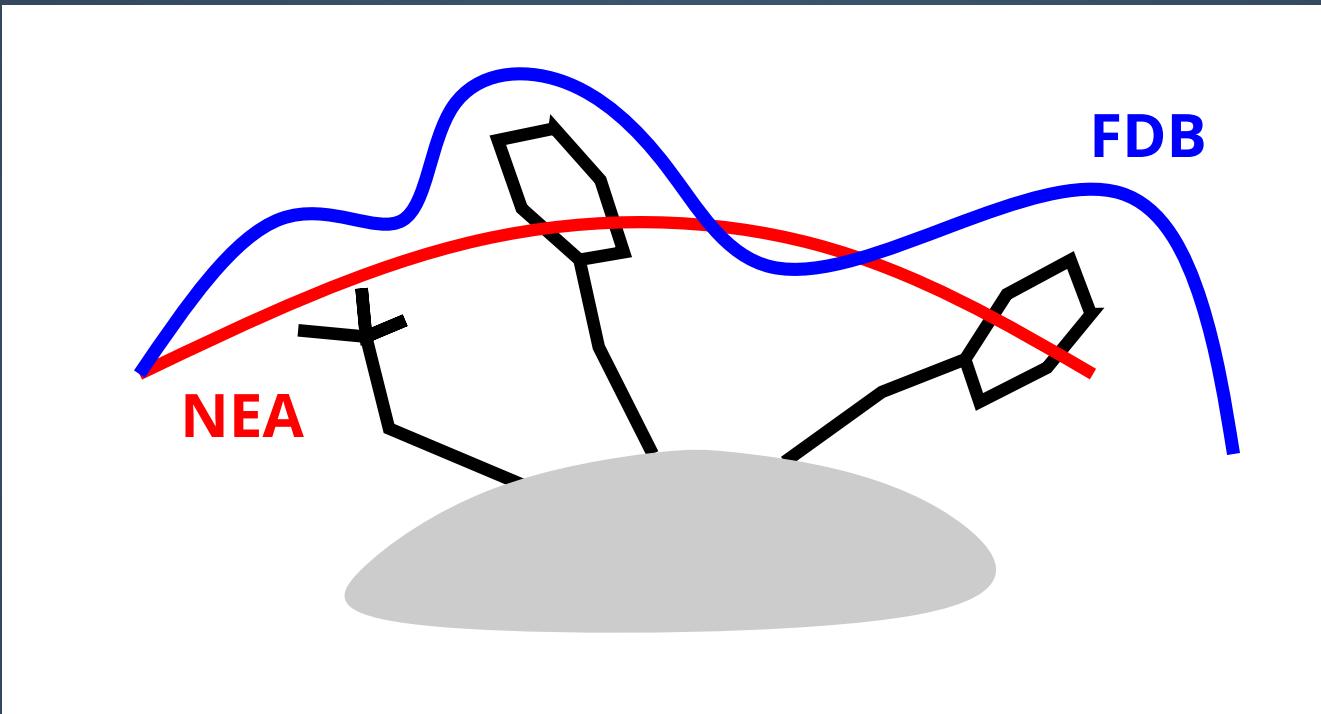
$$C = \sum_{t \in aa} (n_t^{exp} - \langle n(t) \rangle_n)^2$$



# Une nouvelle approximation GB

## Le "Fluctuating Dielectric Boundary" (FDB)

Villa et al (2017)



# Une nouvelle approximation GB

## Le "Fluctuating Dielectric Boundary" (FDB)

Villa et al (2017)

Le terme GB est une somme sur les paires d'atomes de la protéine:

$$E_{GB} = \sum_{i,j}^N g_{ij}(b_i, b_j)$$

On introduit un rayon de solvatation moyen d'un résidu  $I$  :

$$\left( \sum_{i \in I} q_i^2 \right) \frac{1}{B_I} = \sum_{i \in I} \frac{q_i^2}{b_i}$$

Ce qui permet d'avoir:

$$E_{GB} = \sum_{I,J}^N G(B) \quad \text{avec} \quad B = B_I B_J$$

$G$  est fonction de la position relative des chaînes latérales.

On a:

$$G(B) \approx c_1^{IJ} + c_2^{IJ} B + c_3^{IJ} B^2 + c_4^{IJ} B^{-1/2} + c_5^{IJ} B^{-3/2}$$

Les  $c_k^{IJ}$  peuvent être stockés dans la matrice.

# Objectifs et résultats

## L' Exploration dans Proteus

- Le Monte Carlo (mono et multi-marcheurs)
- Comparaison d'algorithmes

## Optimisation et benchmarks

- Les énergies de référence
- Le GB/FDB
- **Protéines PDZ et protocoles**
- Évaluation et comparaison de nos séquences
- Méthode de croissance du noyau hydrophobe

# les protéines

# les homologues

	nombre de positions actives	nombre	E-value	% identité
NHREF	76	62	< 1e-32	67-95
Syntenine	72	85	< 1e-43	85-95
DGL2	82	43	< 1e-41	78-95

# Le protocole

- Les matrices sont calculées avec MM/FDB/SA.
- Les Eref sont calculées avec la méthode du gradient à pas variables.

2 jeux d'énergies selon la position enfuie/exposée

- Le meilleur protocole REMC
- Sélection des 10000 meilleures séquences
- Caractérisation des séquences calculées:  
Superfamily, similarité Pfam (RP55)
- Comparaison avec Rosetta (fixbb)

# Résultats Superfamily et identité

## Proteus

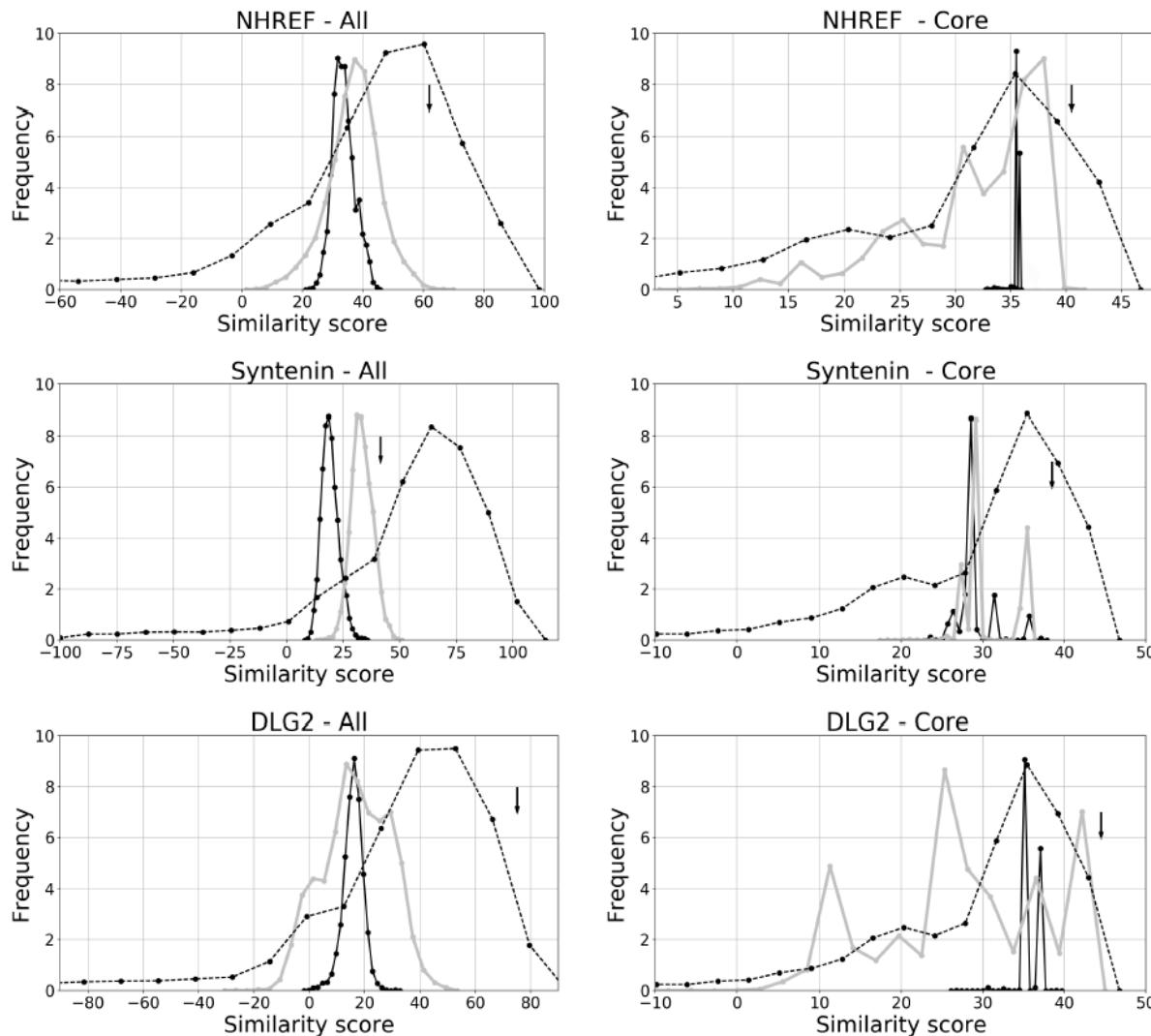
Protéine	identité à la native	E-value Super-famille	succès Super-famille	E-value famille	succès famille
NHREF	24%	$8,54 \cdot 10^{-14}$	100%	$8,94 \cdot 10^{-3}$	100%
Syntenine	31%	$2,85 \cdot 10^{-6}$	100%	$2,69 \cdot 10^{-3}$	100%
DLG2	33%	$3,26 \cdot 10^{-12}$	100%	$1,96 \cdot 10^{-3}$	100%

## Rosetta

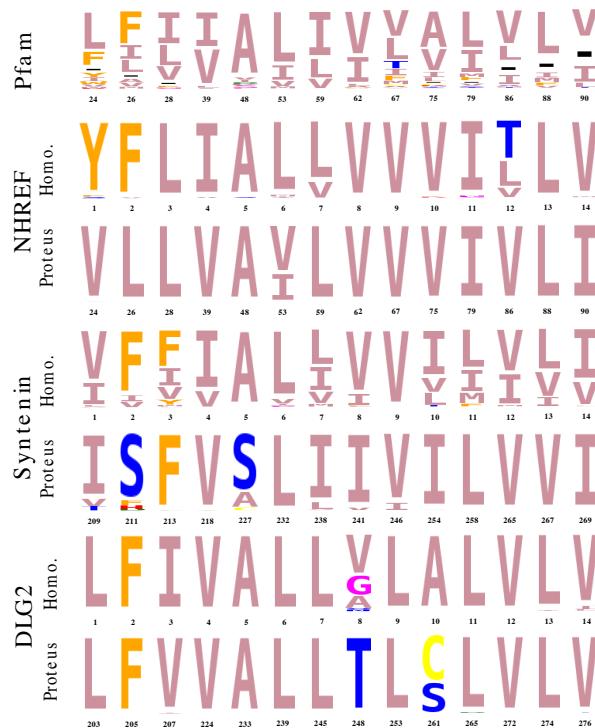
Protéine	identité à la native	E-value Super-famille	succès Super-famille	E-value famille	succès famille
NHREF	35%	$1,3 \cdot 10^{-13}$	100%	$2,2 \cdot 10^{-3}$	100%
Syntenine	38%	$7,3 \cdot 10^{-13}$	100%	$1,8 \cdot 10^{-3}$	100%
DLG2	40%	$1,3 \cdot 10^{-9}$	100%	$9,6 \cdot 10^{-4}$	100%

# Similarité

- Proteus
- Rosetta
- Pfam
- Native



# Séquence Proteus, homologues et Pfam sous forme de logos



# Application: Croissance du noyau hydrophobe

Possibilité de "design" du cœur hydrophobe des domaines PDZ

Tiam1 et Cask sont soumis à plusieurs simulations Proteus.

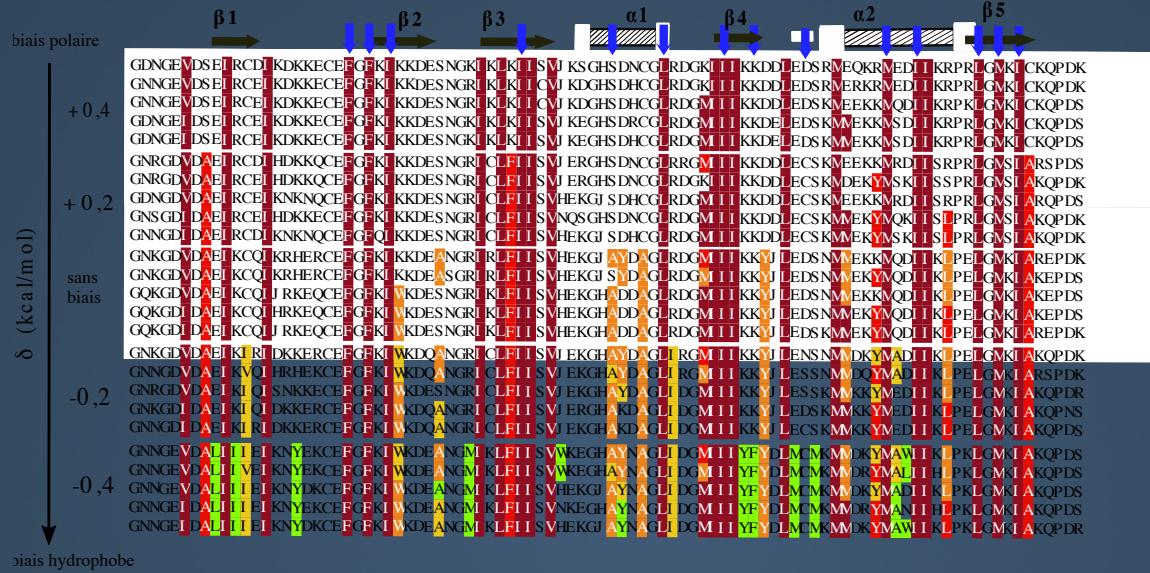
Les simulation sont biaisées graduellement , via les Eref.

Les types hydrophobes sont pénalisés au début , puis progressivement favorisés.

Introduction d'un indice d'hydrophobie:

$$\psi_h = \frac{1}{N} \frac{\delta N}{\delta E}$$

# Croissance du noyau hydrophobe



# Conclusion

1. Introduction du MC/REMC dans proteus:
  - Performant sur les tests tout actif. Proche des meilleurs sur les tests avec espace constraint.
  - Échantillon de qualité
2. Grâce au REMC, une méthode de maximum de vraisemblance des énergies de références est ajoutée.
3. Les performances de Proteus/FDB sont du niveau de Rosetta sur la famille PDZ.
4. Une application illustrative: définition d'un indice d'hydrophobie.

# Annexe

# Entropie

Protéine	Proteus	Rosetta	Pfam "seed"
NHREF	1,38	1,45	3,15
INAD	1,37	1,55	3,06
GRIP	1,33	1,44	3,06
Syntenin	1,39	1,43	3,03
DLG2	1,24	1,57	3,11
PSD95	1,27	1,40	3,15
6 protéines	2,42	2,88	
Cask	1,55	1,65	3,15
Tiam1	1,22	1,57	3,15

# Études de quelques cas

# Séquences au voisinage du GMEC

# 1CKA 10 actifs

## 1M61 10 actifs

# Difficile pour le Monte Carlo

# Difficile pour le MSD

# Une méthode exacte par optimisation combinatoire (Toulbar2)

Equivalence Preserving Transformation (EPT)

pour accélérer le DFBB

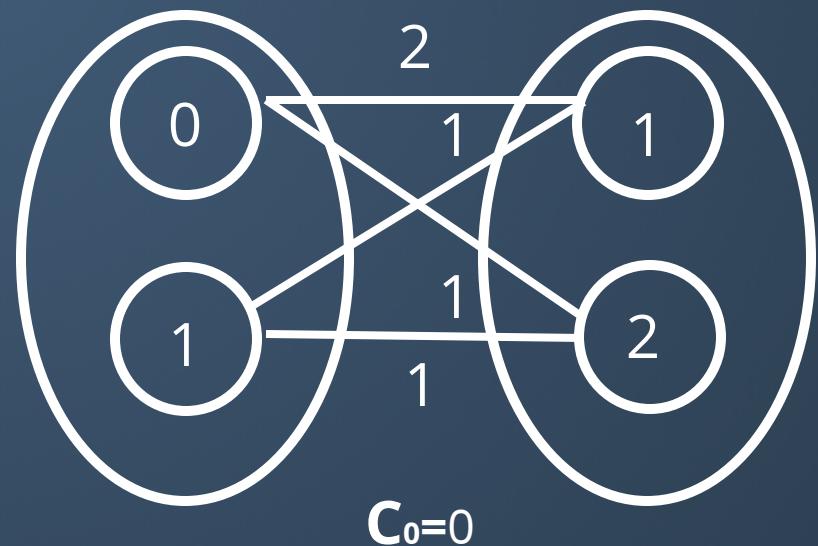
La décomposition par paire de notre fonction d'énergie permet une représentation sous forme d'un réseau de fonctions de coûts

- Une interaction entre acides aminés  $\Leftrightarrow$  une arête du réseau
- Une énergie d'un rotamère  $\Leftrightarrow$  un nœud du réseau

Deux transformations de base:

- la projection
- la distribution

"Dead-End Elimination" en complément



# Une méthode exacte par optimisation combinatoire (Toulbar2)

Equivalence Preserving Transformation (EPT)

pour accélérer le DFBB

La décomposition par paire de notre fonction d'énergie permet une représentation sous forme d'un réseau de fonctions de coûts

- Une interaction entre acides aminés  $\Leftrightarrow$  une arête du réseau
- Une énergie d'un rotamère  $\Leftrightarrow$  un nœud du réseau

Deux transformations de base:

- **la projection**
- la distribution

"Dead-End Elimination" en complément

