

# Proteus: le programme et ses fonctionnalités

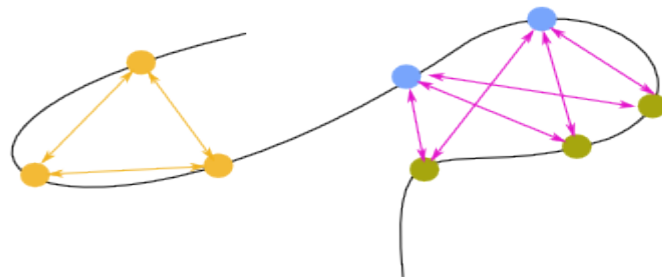
Proteus est un programme d'exploration des séquences et conformations à partir de fichiers d'énergies

La nouvelle version est écrite en C, avec une extension des fonctionnalités et une refonte du système de configuration.

# Énergies, stabilités et affinités.

position	nom de l'acide aminé	nom une lettre	numéro du rotamère	énergie de référence	énergie de Coulomb	autres énergies	énergie surfacique
489	ALA	A	1	-10.894700	-0.128508	0	3.230923E-02
489	ASP	D	1	-19.499700	-0.347171	0.240545	-8.92156
489	ASP	D	2	-19.499700	4.864217E-02	0.589668	-7.64253
489	ASP	D	3	-19.499700	-1.71946	0.357956	-7.90181
489	ASP	D	4	-19.499700	5.546893E-02	-0.971609	-7.62914
489	ASP	D	5	-19.499700	-2.12676	-0.596958	-7.42771
489	ASP	N	1	-17.001400	-0.492971	-0.520301	-5.98663
489	ASP	N	2	-17.001400	-0.596737	-0.549191	-5.64477

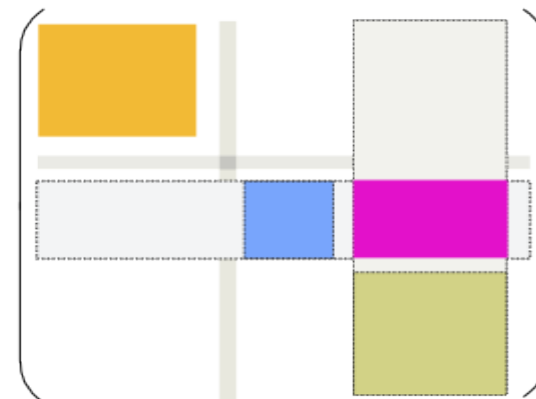
le fichier backbone



Le système

position i	type d'acide aminé de i	position j	type d'acide aminé de j	Les énergies
490	HIS	489	ALA	
1	1			-0.351215
2	1			-0.340499
4	1			-0.112336
7	1			-0.189721
8	1			-0.115124
9	1			-0.167919
490	HIS	489	ASP	
1	1			-0.45112
1	2			-0.561025
1	4			-2.42051
1	5			-3.24513
2	1			-0.438257
2	3			-0.591824
				0.107382
				-2.779489E-02
				0.264577
				0.824753
				5.777122E-02
				0.34835
				0.134858
				0.188942
				0.275176
				5.08406
				5.07537
				0.664548
				0.888732

fichier d'énergies de paires



La matrice

# L'heuristique

Nous utilisons une heuristique type "évolution dirigée", pour trouver rapidement des solutions réalistes.

Le principe de base:

Lors d'un cycle heuristique,

Le programme parcourt la séquence position par position

A chaque position, il effectue un ensemble de mutations et de changements de rotamères.

Après chaque modification, il teste si la nouvelle séquence est meilleur que l'ancienne.

La nouvelle version permet d'optimiser selon l'affinité entre parties du système et de considérer des états différents du système.

# les groupes

- L'utilisateur doit définir des groupes.
- Un groupe simple représente une partie du système.
- un groupe dupliqué représente une deuxième conformation des rotamères d'une partie du système. Un groupe dupliqué peut avoir son jeu d'énergies propre.
- Il y a dans le code, un tableau de groupes et un tableau 2D « des rotamères » courants. Le tableau de groupes est utilisé pour parcourir la «conformation ».

G1	G2	G3	G4	G5
Position: P1 Taille: 3 Type: Simple	Position: P2 Taille: 2 Type: Simple	Position: P3 Taille: 3 Type: Simple	Position: P2 Taille: 2 Type: Dupliqué de Groupe2	Position: P5 Taille: 3 Type: Dupliqué de Groupe3

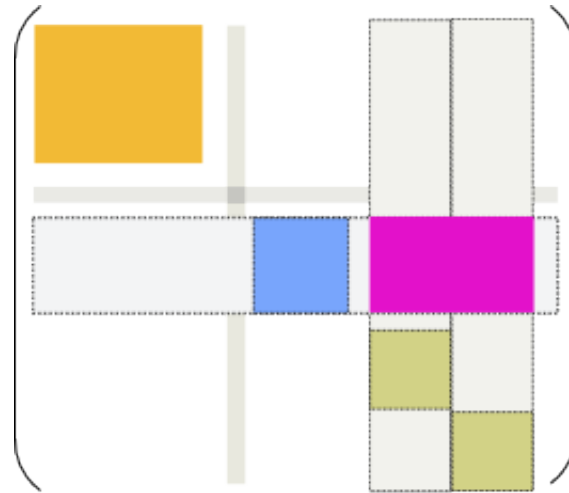
[illegible]

# Matrice d'énergies pour les groupes

- Le programme construit une matrice d'énergies pour les groupes déclarés, fonction de la conformation C:

$E_g(C)$ .

- L'optimisation de Proteus c'est maximiser une fonction F de  $E_g(C)$ .



$E_g(C) =$

$$\begin{pmatrix} E_{1,1} & E_{1,2} & E_{1,3} & E_{1,4} & E_{1,5} \\ E_{1,2} & E_{2,2} & E_{2,3} & 0 & E_{2,5} \\ E_{1,3} & E_{2,3} & E_{3,3} & E_{3,4} & 0 \\ E_{1,4} & 0 & E_{3,4} & E_{4,4} & E_{4,5} \\ E_{1,5} & E_{2,5} & 0 & E_{4,5} & E_{5,5} \end{pmatrix}$$

# l'optimisation

- L'utilisateur doit définir La fonction d'optimisation F par une formule.
- La syntaxe de la formule est composée des éléments suivants:
  - Énergie d'un groupe, représentée par le nom du groupe.
  - Une affinité, entre G1 et G2, représentée par G1~G2.
  - une fonction de maximisation m(), une fonction de seuil t().
  - Les combinaisons linéaires sont possibles.

Exemple :

$$0.2m(0.5G1\sim G2+0.5G2) \ 0.8t(G1)>20$$

Ce qui fixe F comme:

$$F(E_g)= 0.2(0.5E_{1,2}+0.5E_{2,2})+0.8 \min(20,E_{1,1})$$

- Proteus parcourt un ensemble de «conformations».  
Une nouvelle conformation Cn est meilleur que Cp  
Si  $F(E_g(Cn)) > F(E_g(Cp))$ .

# Champ Moyen

- Le principe:

Dans d'un système de N particules en interactions,  
le champ que subit une particule est approximé par la moyenne  
des champs créés par les autres particules.

- On le transpose ainsi:

Pour i,j deux rotamères à deux positions différentes:

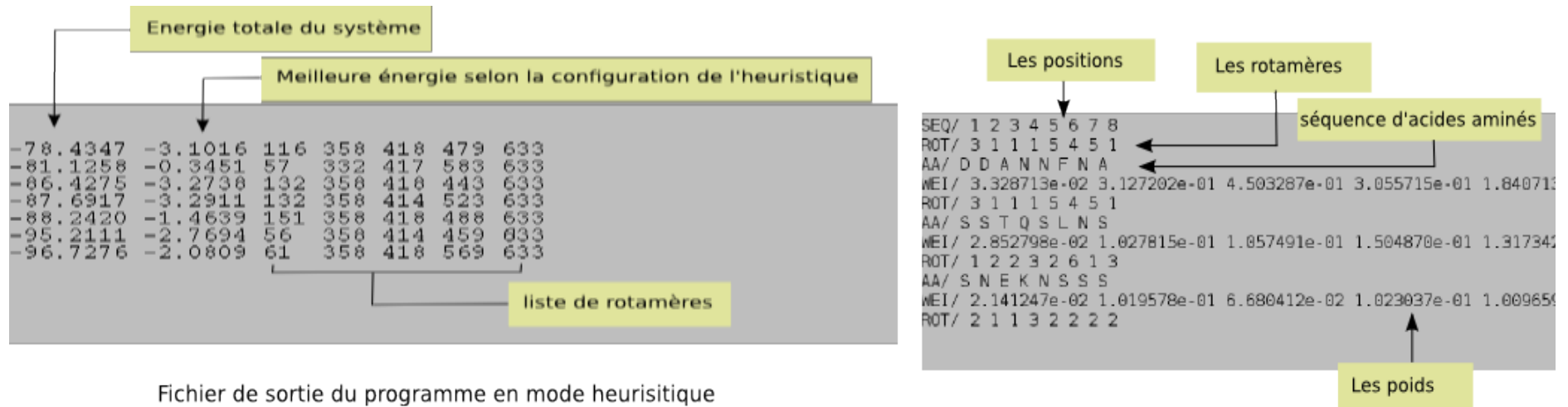
$$E(i) = E(i,i) + \sum_{j \neq i} E(i,j) P(j)$$

avec P() la probabilité de Boltzmann:  $P(i) = \exp(-E(i)/RT)$ .

- Les E(i) et les P(i) sont estimés de façon itératives en partant de poids fixés et en utilisant un mélange:

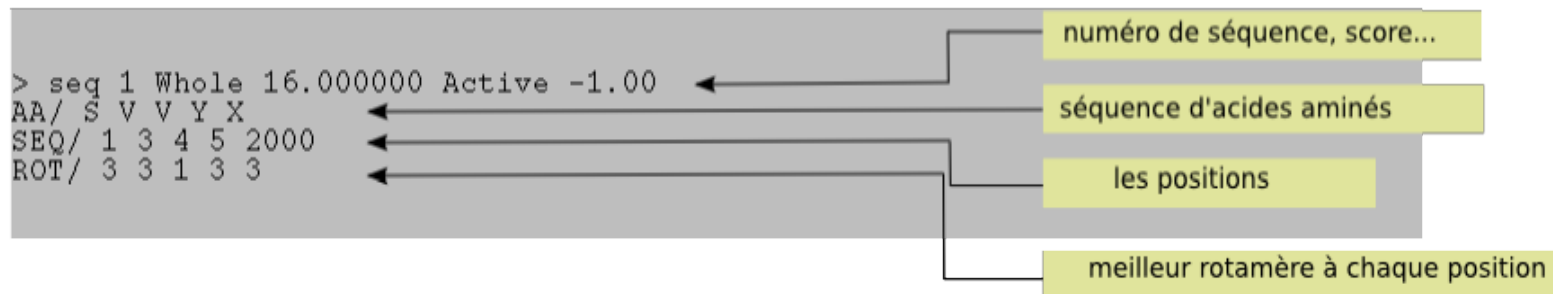
$$P_{n+1}(i) = \lambda P_n(i) + (1-\lambda)P_{n-1}(i)$$

# Le post-traitement



Fichier de sortie du programme en mode heuristique

Fichier de sortie du programme en mode Champ moyen



Fichier de sortie du programme en mode post-traitement



# La configuration, un système de balise

## -a- Choisir le mode de traitement

<MODE> HEURISTIC MEANFIELD POSTPROCESS

## -b- Paramétrer le modèle utilisé

<Dielectric\_Constant>,<Surf\_Ener\_Factor>,<Temperature>,<Lambda\_Parameter> ,<Initial\_Weights>

## -c- Initialiser les conformations de départ

<Initial\_Rotamer\_Conformations>

125 358 421 479 630

126 358 421 479 630

</Initial\_Rotamer\_Conformations>

ou

<Rseed\_Definition>

# La configuration

-d- Limiter l'espace de recherche défini par le fichier backbone:

Une ligne par position où l'on veut limiter la recherche à certains aa ou certains rotamères d'aa:

<Rotamer\_Space\_Restrictions>

489 LYS TRP{1,3,7}

491 ALA GLU

</Rotamer\_Space\_Restrictions>

-e- Définir la fonction d'optimisation

<Group\_Definition> <Heuristic\_Configuration>

-f- Contrôler le déroulement de la recherche

<Optimization\_Cycle\_Number>

<Heuristic\_Cycle\_Number>

# Pour l'utiliser

Dans /bioinfo/proteus , il y a

- Le source, un exécutable
- Une documentation (description des formats de fichiers, des balises de configuration)
- Un modèle de fichier de configuration
- Une douzaine d'exemples (fichiers d'énergies, de configurations, de résultats) des différentes possibilités d'utilisation

# Backbone flexible

- Déclarer les backbones comme des groupes simples:

```
<Group_Definition>
```

```
PROT 1 30
```

```
DUP1 101 130
```

```
DUP2 201 230
```

```
</Group_Definition>
```

- Permettre de définir une compétition entre groupes dans la formule, avec une syntaxe du style:

```
m(PROT|DUP1|DUP2)
```

- modifier l'optimisation pour que la fonction  $F(E_g)$  retourne un tableau d'énergies.
- Après l'optimisation, déterminer le vainqueur de la compétition
- Adapter la boucle d'impression

# Exclusion

- Définir des patterns dans le fichier de configuration:

```
<exclude>
```

```
55 ALA * * * * PHE
```

```
55 PHE * * * * ALA
```

```
80 LYS TRP * ARG * * ALA
```

```
</exclude>
```

- remplacer la ligne

```
for (j = 1 ; j <= res_color_nb[i]; ++j){ //loop over aa
```

```
par quelque chose comme:
```

```
while (next_sequence()){
```

```
next_sequence matche les patterns.
```