

NOM : DUBOIS	Prénoms : Adrien, Sébastien, Pierre
Classe : MP3	
Lycée : Faidherbe	Numéro de candidat : 51771
Ville : Lille	

Concours auxquels vous êtes admissible, dans la banque MP inter-ENS (les indiquer par une croix) :

ENS Cachan	MP - Option MP		MP - Option MPI	
	Informatique			
ENS Lyon	MP - Option MP		MP - Option MPI	
	Informatique - Option M	X	Informatique - Option P	
ENS Rennes	MP - Option MP		MP - Option MPI	
	Informatique	X		
ENS Paris	MP - Option MP		MP - Option MPI	
	Informatique			

Matière dominante du TIPE (la sélectionner d'une croix inscrite dans la case correspondante) :

Informatique	X	Mathématiques		Physique	
--------------	---	---------------	--	----------	--

Titre du TIPE : Application de méthodes de recherche d'isomorphismes de graphes à la comparaison de structures de protéines

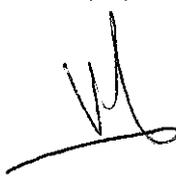
Nombre de pages (à indiquer dans les cases ci-dessous) :

Texte	4, 5	Illustration	2, 5	Bibliographie	1
-------	------	--------------	------	---------------	---

Attention, les illustrations doivent figurer dans le corps du texte et non en fin du document !

Résumé ou descriptif succinct du TIPE (6 lignes, maximum) :

L'objectif est de comparer les structures de protéines lorsqu'elles ont formé leur liaisons et positions dans l'espace. Un outil de comparaison permet de mesurer la précision des prédictions informatiques de la structure à partir d'une séquence d'acides aminés, et ainsi remplacer les méthodes expérimentales longues et coûteuses. La connaissance de ces structures permet, par exemple, de concevoir des médicaments.

A Lille	Signature du professeur responsable de la classe préparatoire dans la discipline	Cachet de l'établissement
Le 16/06/2022		Lycée FAIDHERBE 9, rue Armand Carrel B.P. 767 - 59034 LILLE Cedex Tél. : 03.20.60.5000 Fax : 03.20.60.5005
Signature du (de la) candidat(e) 	La signature du professeur responsable et le tampon de l'établissement ne sont pas indispensables pour les candidats libres (hors CPGE).	

Application de méthodes de recherche d'isomorphismes de graphes à la comparaison des structures de protéines

Adrien Dubois - N° Candidat : 51771

16 juin 2022

Table des matières

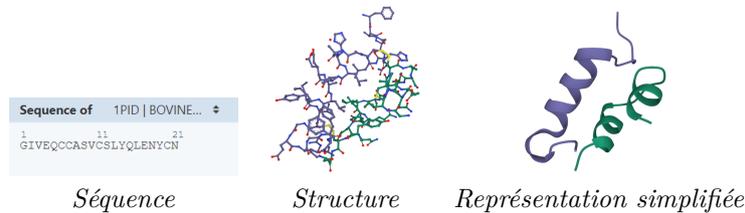
1	Contexte, données à traiter et objectif	2
1.1	Contexte, objectif	2
1.2	Format PDB [3]	2
2	Comparaison de branches (pas de ramifications)	3
2.1	Branche	3
2.2	Coefficient de comparaison	3
2.3	Valeur du coefficient sur des exemples	3
3	Isomorphisme de protéines	3
3.1	Isomorphisme de graphes	3
3.2	Force brute	4
3.3	Tri des sommets par invariants	4
4	Isomorphisme canonique de McKay [2]	4
4.1	Tri équitable [1]	4
4.2	Arbre de recherche [1]	5
4.3	Isomorphisme canonique [1]	6
5	Plus grand sous-graphe isomorphe commun	6
5.1	Choix de la méthode utilisée pour le test d'isomorphisme sur deux sous-graphes	6
5.2	Itération sur les sous-graphes	7
5.3	Cas d'égalité	7
6	Comparaison des positions de structures isomorphes	7
6.1	Application du coefficient des branches sur les structures	7
7	Comparaison de protéines	7
7.1	Coefficient de comparaison	7
7.2	Exemples de coefficients de comparaison de protéines	7

1 Contexte, données à traiter et objectif

1.1 Contexte, objectif

La compétition Casp est un concours de prédiction informatique de structures de protéines à partir de séquences d'acides aminés. Les modèles proposés sont déjà connus, il faut alors comparer les structures informatiques et expérimentales. Un moyen de comparaison permettrait d'affiner la précision des prédictions et ainsi remplacer les méthodes de détermination expérimentales longues et coûteuses par des méthodes informatiques.

Finalement, la structure d'une protéine permet une meilleure compréhension de la protéine ; par exemple est nécessaire pour la fabrication de médicaments.



source : Protein Data Bank, <https://www.rcsb.org/3d-view/1B0Q> [3]

Objectif : proposer une méthode de comparaison de structures

1.2 Format PDB [3]

Voci un fichier PDB :

```
ATOM 171 HG13 VAL A 10 -7.189 -6.908 -5.228 1.00 0.00 H
HETATM 172 N NH2 A 11 -3.913 -3.201 -4.868 1.00 0.00 N
HETATM 173 HN1 NH2 A 11 -3.068 -3.568 -5.283 1.00 0.00 H
HETATM 174 HN2 NH2 A 11 -3.878 -2.361 -4.308 1.00 0.00 H
TER 175 NH2 A 11
HETATM 176 RE RE A 182 2.230 -1.164 0.585 1.00 0.00 RE
CONNECT 1 2 3 7
CONNECT 2 1
CONNECT 3 1 4 5 6
```

FIG. 1 : Lignes d'un fichier PDB

A partir de ce fichier on extrait :

- les atomes
- le type (carbone, hydrogène, etc)
- la position dans l'espace des atomes
- les liaisons (attention, à cause de considérations biologiques non prises en compte peu de liaisons sont lues)

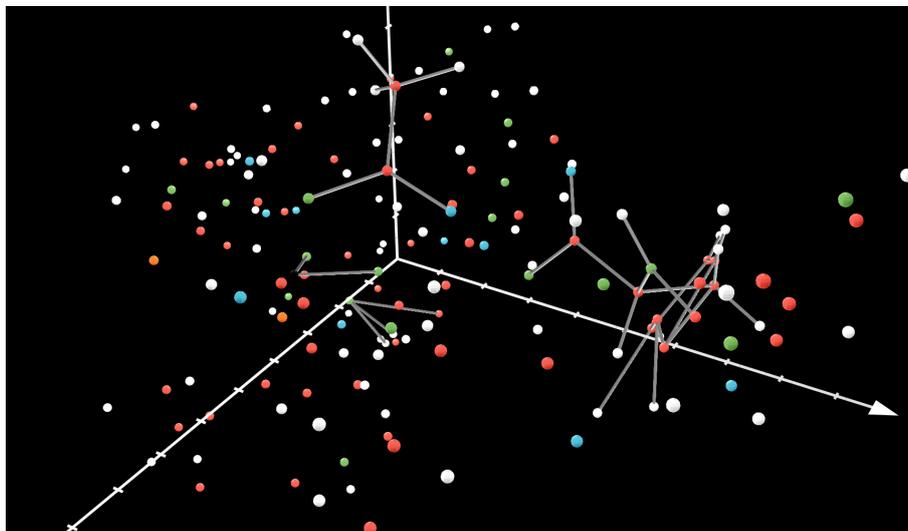


FIG. 2 : Affichage du PDB à l'aide de la bibliothèque python manim

2 Comparaison de branches (pas de ramifications)

2.1 Branche

Une branche $B_1 = [M_1, \dots, M_n]$ est une suite de points ordonnés dont les liaisons sont de proche en proche (liaison de M_1 à M_2 , M_2 à M_3 , etc)

2.2 Coefficient de comparaison

L'idée est d'associer à deux branches données de même taille un coefficient de proximité $R \in [0, 1]$, deux paramètres sont considérés :

- l'écart moyen des longueurs des liaisons
- les angles que forment deux liaisons situées au même niveau des branches

Soit $B_1 = [M_1, \dots, M_n]$ et $B_2 = [N_1, \dots, N_n]$ des branches. On pose $\forall i \in \llbracket 1, n-1 \rrbracket$:

$$l_i = M_i M_{i+1}, l'_i = N_i N_{i+1}, d_i = \max(l_i, l'_i), \text{ et } \theta_i = \text{angle}(\overrightarrow{M_i M_{i+1}}, \overrightarrow{N_i N_{i+1}})$$

$$C_{dist} = \frac{\sum_{i=0}^{n-1} |\sin(\frac{\theta_i}{2})| d_i}{\sum_{i=0}^{n-1} d_i} \quad C_{angle} = \frac{\sum_{i=0}^{n-1} |l'_i - l_i|}{\sum_{i=0}^{n-1} d_i}$$

$$R_{dist} = \frac{1}{1 + C_{dist}} \quad R_{angle} = \frac{1}{1 + C_{angle}}$$

Finalement, on définit :

$$R = \frac{R_{dist} + R_{angle}}{2}$$

2.3 Valeur du coefficient sur des exemples

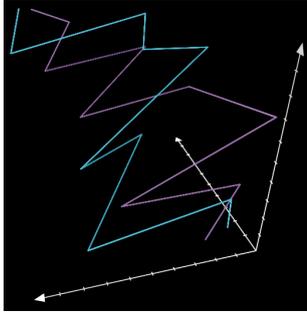


FIG. 3 : Branche A

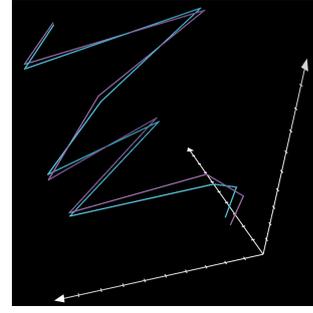


FIG. 4 : Branche B

$$R_{dist} = 0.59 \quad R_{angle} = 0.68$$

$$R = 0.64$$

$$R_{dist} = 0.94 \quad R_{angle} = 0.97$$

$$R = 0.95$$

3 Isomorphisme de protéines

3.1 Isomorphisme de graphes

Définition : Soit $A_g = \{g_1, \dots, g_n\}$ et $A_h = \{h_1, \dots, h_n\}$ des ensembles
Soit $G = A_g \times S_g$ et $H = A_h \times S_h$ deux graphes où $S_g, S_h \subseteq [n] \times [n]$
(ainsi, l'arête (g_i, g_j) est dans $G \iff (i, j) \in S_g$) :

$$G \cong H \text{ si et seulement si } \exists \sigma \in \Sigma_n, S_g = S_h^\sigma \quad (1)$$

où $S_h^\sigma := \{(\sigma(i), \sigma(j)), \exists (i, j) \in [1, n]^2, (i, j) \in S_h\}$

3.2 Force brute

Itération sur $n!$ permutations, on test si H permuté est le même graphe que G

3.3 Tri des sommets par invariants

On crée une partition de $\llbracket 1, n \rrbracket$ des sommets $(V_1 | \dots | V_t)$ pour G et $(W_1 | \dots | W_t)$ pour H tels que $\forall i \in \llbracket 1, t \rrbracket$, $V_i = (v_{i,1} \dots v_{i,|V_i|})$ et $W_i = (w_{i,1} \dots w_{i,|W_i|})$ alors :

$$G \cong H \iff \exists (\sigma_i)_{i \in \llbracket 1, t \rrbracket} \in \Sigma_{|V_1|} \times \dots \times \Sigma_{|V_t|}, G = H^\sigma \text{ avec } \sigma : \begin{pmatrix} \forall j \leq |W_1|, w_{1,j} \mapsto w_{1,\sigma_1(j)} \\ \dots \\ \forall j \leq |W_t|, w_{t,j} \mapsto w_{t,\sigma_t(j)} \end{pmatrix} \quad (2)$$

Exemple :

$$\begin{aligned} (V_1, V_2) &\rightarrow (1 \ 2 \ 3 \ | \ 4 \ 5 \) \\ (W_1, W_2) &\rightarrow (1 \ 3 \ 5 \ | \ 2 \ 4 \) \end{aligned}$$

σ_1 ¹	σ_2	$(W_1^{\sigma_1} W_2^{\sigma_2})$	$\sigma = \tilde{\sigma}_1 \circ \tilde{\sigma}_2$
(1 2 3) = Id	(1 2) = Id	(1 3 5 2 4)	(1 3 5 2 4)
(3 2 1)	(1 2) = Id	(5 3 1 2 4)	(5 3 1 2 4)
(1 2 3) = Id	(2 1)	(1 3 5 4 2)	(1 3 5 4 2)

G et H sont des graphes à n sommets tous deux **triés canoniquement** :

- types des $t \in \mathbb{N}$ paquets du tri sont dans le **même ordre**
- on suppose les paquets de chacun des tris de **même taille** t_i , $i \in \llbracket 1, t \rrbracket$ **deux à deux** (sinon les graphes seraient trivialement non isomorphes)

	force brute	tri des sommets
procédé d'itération	permutations sur Σ_n	combinaison de permutations sur $(\Sigma_i)_{i \in \llbracket 1, t \rrbracket}$ tel que $\sum_{i=1}^t t_i = n$
nombre maximal d'itérations	$n!$	$\prod_{i=1}^t t_i!$
mise en mémoire nécessaire	$\sum_{k=1}^n k!$ permutations	$\sum_{k=1}^{\max(t_1, \dots, t_t)} k!$ permutations

4 Isomorphisme canonique de McKay [2]

- Idee générale :
- tri équitale des sommets à partir d'un tri
 - informations avec la propagation du degré
 - tri optimal
 - création d'un arbre de recherche de permutations
 - on crée artificiellement de nouveaux tris équitables (fils de l'arbre) en isolant des sommets
 - les feuilles sont des tris ordonnés de parties à un sommet : ce sont des permutations
 - définition d'un ordre total sur les graphes
 - déterminer le plus grand pour cette relation parmi les graphes permutés avec les feuilles de l'arbre
 - on obtient alors l'isomorphisme canonique

4.1 Tri équitale [1]

Utilisation de la propagation du degré :

Tri des sommets des paquets par degré dans les autres paquets pour en faire un nouveau tri, jusqu'à ce qu'il n'y ait plus de simplification possible

Soit $\pi = (1 \mid 3 \ 7 \ 9 \mid 6 \ 8 \mid 2 \ 4 \mid 5)$ un tri des sommets du graphe G
 $= (V_1 \mid V_2 \mid V_3 \mid V_4 \mid V_5)$

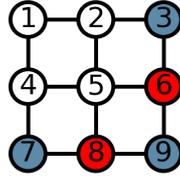


FIG. 5 : Graphe G avec $V_2 = (3 \ 7 \ 9)$ et $V_3 = (6 \ 8)$

Tri de V_2 par degré dans V_3 : $\deg(3, V_3) = \deg(7, V_3) = 1$ et $\deg(9, V_3) = 2$
donc $V_2 = (3 \ 7 \mid 9)$ et $\pi' = (1 \mid 3 \ 7 \mid 9 \mid 6 \ 8 \mid 2 \ 4 \mid 5)$ est plus fin

underlineTri équitable : la propagation de degré ne donne plus de nouveau tri

$$\begin{aligned} & \forall (i, j) \in \llbracket 1, n \rrbracket^2, \text{ tous les sommets de } V_i \text{ ont le même degré dans } V_j \\ \iff & \forall (i, j) \in \llbracket 1, n \rrbracket^2, \forall (v, w) \in V_i^2, \deg(v, V_j) = \deg(w, V_j) \end{aligned}$$

Relation d'ordre partiel sur les partitions Soit $\pi_1 = (V_1 \mid \dots \mid V_t)$ et $\pi_2 = (W_1 \mid \dots \mid W_{t'})$ des partitions de $\llbracket 1, n \rrbracket$

$$\text{alors } \pi_1 \text{ est plus fin que } \pi_2 \iff \begin{cases} \forall V \in \pi_1, \exists W \in \pi_2, V \subseteq W \\ \forall i \leq j, \forall W_k, W_l, (V_i \subseteq W_k \text{ et } V_j \subseteq W_l \Rightarrow k \leq l) \end{cases} \quad (3)$$

Soit π un tri, on note :

$$\boxed{R(\pi) \text{ le tri équitable ordonné le plus fin obtenu à partir de } \pi} \quad (4)$$

4.2 Arbre de recherche [1]

On considère le tri suivant le degré $\pi = (1 \ 3 \ 7 \ 9 \mid 2 \ 4 \ 6 \ 8 \mid 5)$ dans le graphe G (défini ci-dessus).

Principe de l'arbre de recherche :

- racine de l'arbre : $R(\pi)$
- trouver les fils :
 - trouver la première partie V_i d'au moins 2 éléments de π
 - pour $v \in V$, on crée **artificiellement** un nouveau tri équitable :
 $\pi_v = \pi \perp v = R((V_1 \mid \dots \mid \{v\} \mid V_i \setminus \{v\} \mid \dots))$
 - chacun des tris créés est un fils, on réitère jusqu'à obtenir des tris triviaux (paquets de taille 1)

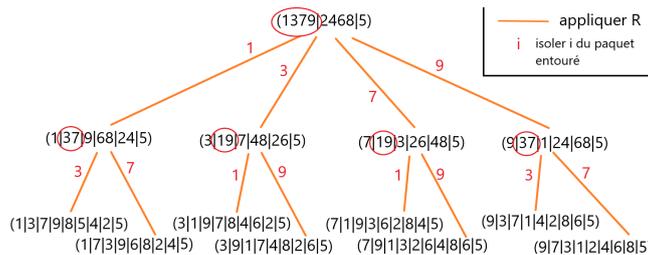


FIG. 6 : Arbre $T(G)$ de racine $\pi = (1 \ 3 \ 7 \ 9 \mid 2 \ 4 \ 6 \ 8 \mid 5)$

4.3 Isomorphisme canonique [1]

Ordre total \preceq sur les graphes :

On pose la fonction $i : G \mapsto i(G)$ telle que :

- $i(G)$ est la séquence binaire $(\mathbf{1}_{(i,j) \in G})$ avec (i, j) dans l'ordre lexicographique
- $G \preceq H$ si et seulement si $i(G) \leq i(H)$ en décimal

On pose alors l'**isomorphisme canonique de McKay** (pour le tri π) :

$$C_M(G) = \max_{\preceq} \{G^\sigma, \sigma \text{ noeud terminal de } T(G) \text{ de racine } \pi\} \quad (5)$$

alors :

$$G \cong H \text{ si et seulement si } C_M(G) = C_M(H) \quad (6)$$

Exemple : pour $\pi = (1 \mid 3 \ 7 \mid 9 \mid 6 \ 8 \mid 2 \ 4 \mid 5)$

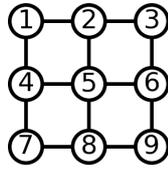


FIG. 7 : Graphe G

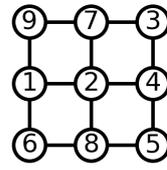


FIG. 8 : Graphe $C_M(G)$

5 Plus grand sous-graphe isomorphe commun

5.1 Choix de la méthode utilisée pour le test d'isomorphisme sur deux sous-graphes

Dans le cas particulier des protéines, le tri équitable (tri le plus fin obtenu par propagation du degré) proposé par McKay est très efficace et produit des paquets de très petite taille. En combinant ce tri et le test d'isomorphisme sur les graphes à sommets triés, on obtient un algorithme très efficace, plus efficace que la méthode complète de McKay.

Exemple :

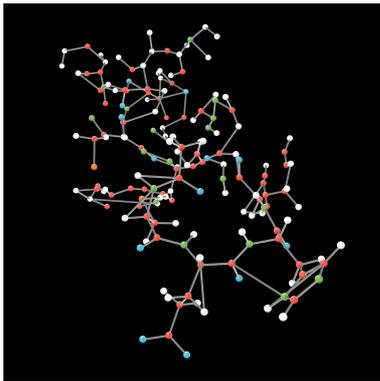


FIG. 9 : Protéine

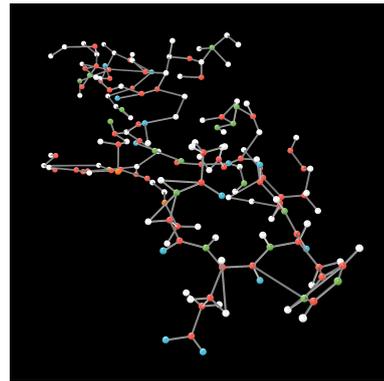


FIG. 10 : Même protéine mais déplacée et numérotation différente des sommets

Temps d'exécution : pour cette protéine de 171 atomes

tri par degré et atomes	0.002 s
propagation des degrés	7.4 s
test d'isomorphisme	0.3 s
temps total d'exécution	7.8 s

5.2 Itération sur les sous-graphes

On cherche la sous-structure de taille maximale de $G_1 = A_1 \times S_1$ et $G_2 = A_2 \times S_2$:

- itération pour k allant de $\min(|S_1|, |S_2|)$ à 1 (sur le nombre d'arêtes du sous-graphe)
- pour un sous-graphe de k arêtes, on itère sur l'ensemble de parties de k arêtes de G_1 formant un graphe connexe
- pour un sous-graphe donné de G_1 de k arêtes, on itère sur l'ensemble de parties de k arêtes de G_2 formant un graphe connexe
- test d'isomorphisme sur les sous-graphes de k arêtes

Complexité en nombre de tests d'isomorphismes : exponentielle, en $O(4^{\min(|S_1|, |S_2|)})$ mais beaucoup de cas sont écartés rapidement, si les sous-graphes ne sont pas connexes ou les tris formés de tailles différentes

NB : on peut trouver plusieurs sous-graphes isomorphes de taille maximale
Comment choisir parmi ces sous-structures isomorphes ?

5.3 Cas d'égalité

L'idée est de comparer comment les structures sont formées dans l'espace.

6 Comparaison des positions de structures isomorphes

6.1 Application du coefficient des branches sur les structures

Il existe une bijection entre les arêtes des structures comparées (on les suppose isomorphes). Il est alors possible d'utiliser le même coefficient de comparaison que pour les branches, après alignement, en ordonnant les arêtes des structures.

7 Comparaison de protéines

7.1 Coefficient de comparaison

L'idée est d'associer à deux protéines données un coefficient de comparaison $C \in [0, 100]$ selon deux paramètres :

- la taille des plus grandes sous-structures isomorphes : $C_{struct} = \frac{\text{taille sous-struct}}{\text{taille moyenne des 2 protéines}}$
- la proximité dans l'espace des deux plus grandes sous-structures isomorphes : C_{prox} est le coefficient de comparaison de branches mais appliqué sur les arêtes des sous-structures isomorphes

alors
$$C = \frac{C_{prox} + C_{struct}}{2}$$

7.2 Exemples de coefficients de comparaison de protéines

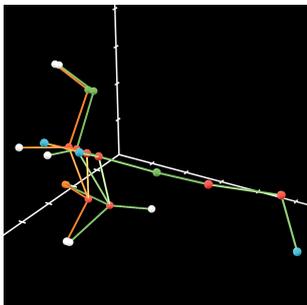


FIG. 11 : Protéines plutôt éloignées

$$C_{prox} = 92.9 \quad C_{struct} = 52$$

$$C = 72$$

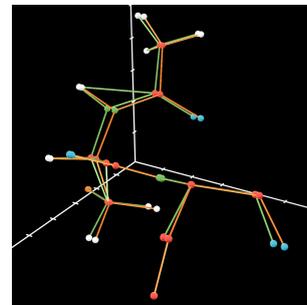


FIG. 12 : Protéines plutôt proches

$$C_{prox} = 97.7 \quad C_{struct} = 93$$

$$C = 95$$

Références

- [1] Stephen G. Hartke and A. J. Radcliffe, McKay's Canonical Graph labeling algorithm, 2008
- [2] Brendan D. McKay, Pratical Graph Isomorphism, 1981
- [3] Base de données PDB, <https://www.rcsb.org/>