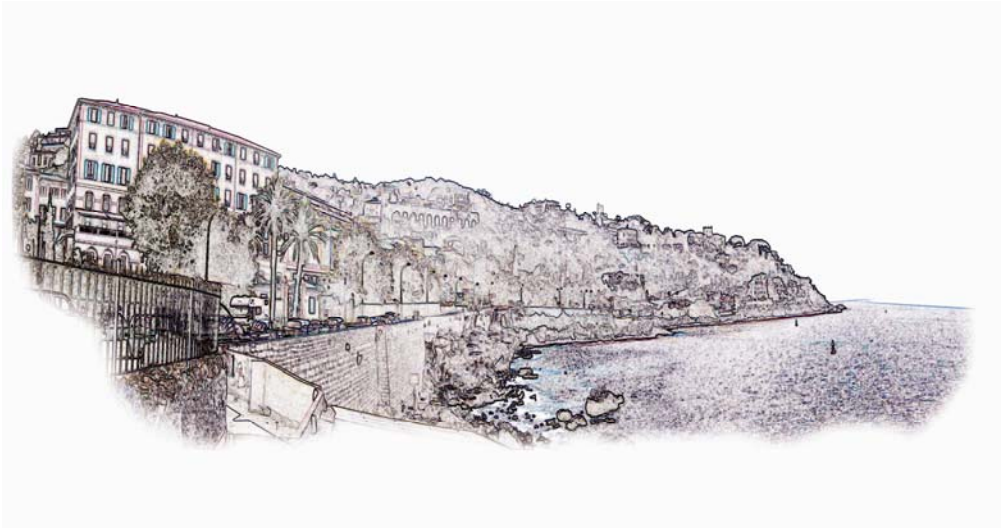
Plate-forme AFIA / Nice, du 30 mai au 3 juin 2005

ATELIER :

Apprentissage automatique et bioinformatique



Florence D'Alche-Buc
Laurent Bréhélin

Sommaire

Extraction et Comparaison entre N-Grammes et Descripteurs Discriminants pour la Classification de Protéines

Faouzi Mhamdi [1], Mondher Maddouri [2] et Mourad Elloumi [3]

Unité de Recherche en Programmation, Algorithmique et Heuristique

[1] Institut Supérieur d'Informatique,
Université d'El Manar, Tunis, Tunisie
02 Rue Abou Rayhan Albayrouni, 2080 Ariana, Tunisie
faouzi.mhamdi@ensi.rnu.tn
[2] Institut National des Sciences Appliquées et de Technologie,
Université 7 Novembre de Carthage, Centre Urbain Nord,
BP. 676, 1080 Tunis, Tunisie
mondher.maddouri@fst.rnu.tn
[3] Faculté des Sciences Economiques et de Gestion de Tunis,
Université d'El Manar, Tunis,
Campus Universitaire, 1060 Tunis, Tunisie
mourad.elloumi@fsegt.rnu.tn

Résumé : La classification des protéines est un domaine d'actualité. Sachant qu'une protéine peut être représentée par différentes manières : sous forme de structures primaires, secondaires ou tertiaires. La classification de protéines est fortement dépendante de l'extraction des caractéristiques servant à une représentation attribut/valeur. Dans ce travail, on a comparé deux approches d'extraction de caractéristiques basées sur l'extraction de sous séquences d'acides aminé (en structure primaire). La première concerne les Descripteurs Discriminants. La deuxième est celle des n-grammes. Nous avons mené une étude comparative théorique et empirique entre ces deux approches.

Mots clés : Classification de Protéines, Fouille de Données, N-Grammes, Descripteurs Discriminants

Introduction

La forte croissance des données biologiques rend indispensable l'utilisation d'outils informatique pour leur manipulation et leur traitement. Dans cet article, nous nous appuyons sur le cadre générique d'ECD (Extraction de Connaissances à partir de Données), pour classer des protéines à partir de leurs structures primaires. Le processus d'ECD se compose principalement de trois phases : prétraitement des données, fouille de données et post-traitement des connaissances découvertes (Fayyad *et al.,* 1996).

La structure primaire d'une protéine est décrite par une chaîne de caractères représentant des acides aminés, il y a 20 acides aminés possibles. La figure 1 présente un exemple de fichier décrivant quelques protéines.

La plupart des méthodes de fouille de données, nécessitent la représentation des données d'apprentissage en tableau attributs/valeurs. Où les attributs représentent les variables prédictives. Ces méthodes sont incapables de traiter directement des données non structurées. Il est donc nécessaire de transformer la représentation en structure primaires des protéines en un tableau attributs/valeurs. Un enjeu important d'ECD est donc l'extraction des « bons » caractéristiques avec lesquelles on construit ce tableau. Ces caractéristiques prédictives seront extraites à partir de la description initiale des protéines. C'est l'étape d'extraction de caractéristiques de la phase de prétraitement des données (Fayyad *et al.,* 1996).

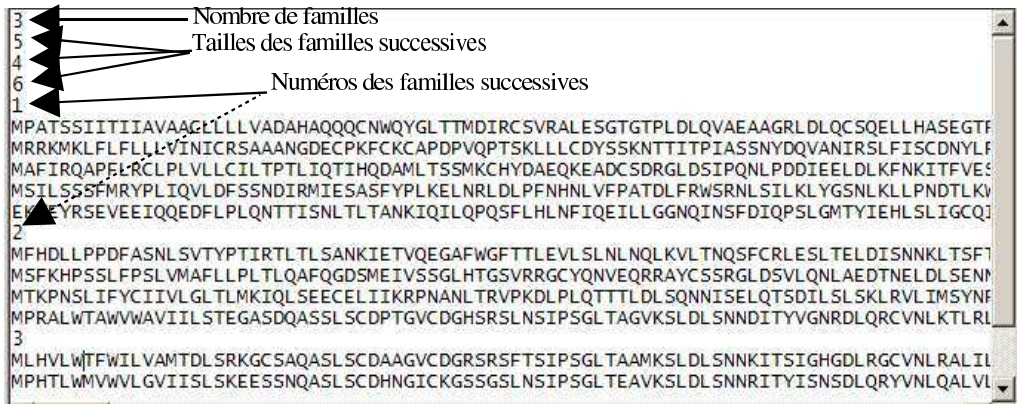


Fig. 1 - Fichier de Familles de protéines

Dans notre cas, le tableau d'apprentissage est un tableau booléen, où chaque ligne représente une séquence et chaque colonne représente une caractéristique. Le contenu du tableau montre si une caractéristique appartient à une séquence (1) ou non (0). La figure 2 illustre ce principe.

Ensemble de séquences

Caractéristiques (sous-séquence)

	MPA	PAT	ATS	TSS	SSI	SII	III	ITI	TII	IIA	IAV	AVA	VAA	AAC
Seq0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Seq1	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Seq2	0	1	0	1	0	0	1	0	1	0	0	0	0	0
Seq3	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Seq4	0	1	0	0	0	1	0	0	1	1	0	0	0	0
Seq5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Seq6	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Seq7	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Seq8	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Fig. 2 - Tableau booléen d'apprentissage

Dans cet article, on étudie l'extraction de caractéristiques à partir de structures primaires de protéines. On présente deux approches : l'approche des n-grammes (Miller *et al.*, 1999) et l'approche des descripteurs discriminants (DDs) (Maddouri & Elloumi, 2002). Et ce, dans le but de comparer les résultats de ces deux approches pour la classification de protéines et d'identifier leurs avantages et leurs inconvénients. Le schéma général du processus de classification est résumé dans la figure 3.

Une étude empirique a été menée pour identifier les caractéristiques qui discriminent le mieux une famille de protéines par rapport à une autre. Ces descripteurs sont comparés selon leur taux de mauvaises classifications (ou taux d'erreur). Pour l'évaluation du taux d'erreurs, on a utilisé la méthode de validation croisée (Hastie *et al.*, 2001). Tandis que pour la classification on a utilisé les méthodes du plus proche voisin (1-PPV) et d'arbres de décisions (C4.5) (Lefébure & Venturi, 2001).

Ce papier est organisé comme suit. Dans la section 2, on présente l'approche DDs d'extraction de caractéristiques. Dans la section 3, on présente l'approche n-grammes d'extraction de caractéristiques. Dans la section 4, on présente les expérimentations qu'on a réalisées et leurs résultats. Enfin, on conclu en discutant les avantages et les inconvénients de chacune des deux approches.

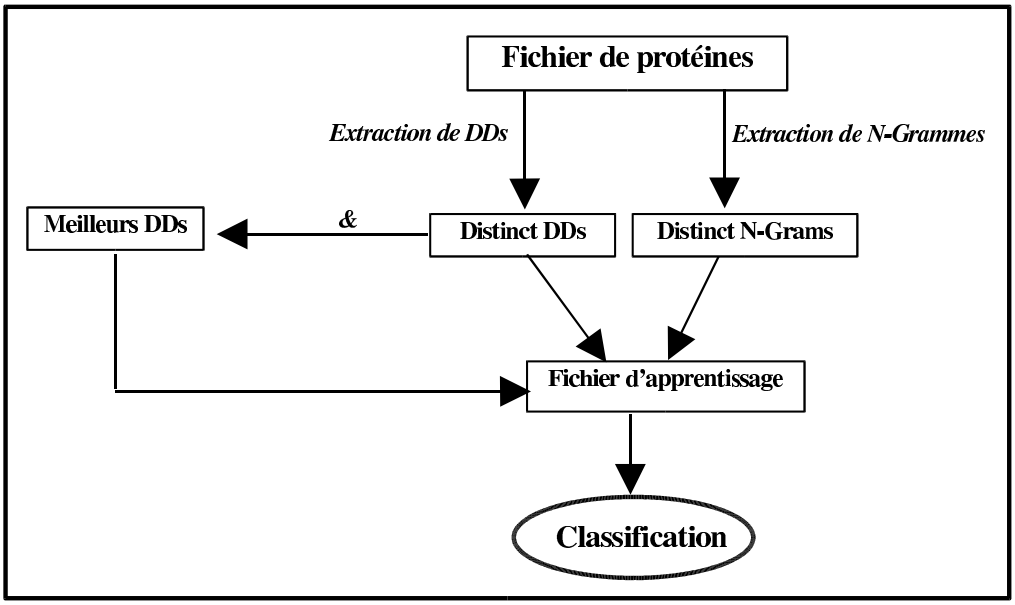


Fig. 3 - Approches d'extraction des DDs et des n-grammes

L'approche des descripteurs discriminants

Dans cette section, on va détailler l'approche des descripteurs discriminants (DDs). Les DDs sont des sous séquences composées d'acides aminés. L'approche des DDs permet de trouver les DDs qui discriminent une famille de protéines par rapport à d'autres. Ces DDs permettent, par la suite, d'affecter une nouvelle séquence de protéine dans la bonne famille.

L'extraction des DDs est basée sur une adaptation de l'algorithme KMR (Karp, Miller, Rosenberg) (Karp *et al.*, 1972). L'algorithme KMR se base sur un théorème associant une relation d'équivalence à chaque longueur de mot et définissant une relation permettant de déduire les mots de longueur n+1 à partir de ceux de longueur n.

Soit W une chaîne de longueur L, et soient i, j, et k trois entiers tels qu'on a :

- $1 \leq k \leq L$.
- $1 \leq i \leq j \leq L - k + 1$.

On dit que les positions i, et j appartiennent à la même classe d'équivalence pour la relation d'équivalence E_k, si et seulement si , la propriété suivante est vérifiée :

- $W_{i,i+k-1} = W_{j,j+k-1}$.

On dit aussi que les positions i et j sont k-équivalentes.

Une relation d'équivalence E_k , $1 \leq k \leq L$ peut être représentée par un vecteur V_k :

- $V_k = (V_{k,1}, V_{k,2}, \ldots, V_{k,L-k+1})$ où chaque composante $V_{k,i}$, $1 \leq i \leq L - k + 1$, de ce vecteur représente le numéro de la classe d'équivalence à laquelle appartient la position i pour la relation d'équivalence E_k.

Théorème : Soit W une chaîne de longueur L et soient a, b, i, et j quatre entiers tels qu'on a :

- $b \leq a$.
- $1 \leq a+b \leq L$.
- $1 \leq i < j \leq L - a - b + 1$.

On a alors $\mathbf{i\ E_{a+b}\ j}$ si et seulement si, on a $\mathbf{i\ E_a\ j}$ et $\mathbf{(i + b)\ E_a\ (j + b)}$.

L'algorithme donc commence par l'extraction des sous séquences de taille 1 (les caractères isolés), ainsi que leurs relations d'équivalences (le vecteur V_1, avec a =1). Puis, conformément au théorème précédent, on cherche les DDs de taille 2 en faisant des concaténations avec les DDs construit précédemment (on considère b=1). Ce principe sera itéré jusqu'à identifier tous les DDs possibles.

L'algorithme doit s'assurer que les DDs sélectionnées sont minimaux. Un DD minimal ne contient pas d'autre DDs. Voici un exemple qui explique la notion du DD minimal. Soient DD_1 et DD_2 deux sous séquence d'acides aminé, tel que DD_1 = « A**DCG**T » et DD_2 = « **DCG** ». L'algorithme ne peut pas considérer DD_1 et DD_2 comme deux descripteurs différents d'un même fichier, car DD_2 est inclus dans DD_1, d'où DD_1 n'est pas minimal.

L'algorithme doit s'assurer que les DDs sélectionnées sont discriminants. L'algorithme d'extraction de DDs possède deux paramètres α et β. La valeur de α signifie que les DDs qui seront extrait doivent être présent au moins α fois dans la première famille. Alors que la valeur de β signifie que les DDs qui seront extrait doivent être présent au plus β fois dans les autres familles. Si $\alpha = 0$ et $\beta = 0$, aucune contrainte de discrimination n'est exigée. On obtient, ainsi, le nombre total de DDs. Dans le cas où α et β sont différents de 0 le nombre de DDs diminue considérablement. Cependant, on peut considérer que les paramètres α et β sont deux paramètres de filtrage qui servent à sélectionner les meilleurs DDs.

La complexité de l'algorithme est limitée par la construction des relations d'équivalence E_{a+b} en fonction de E_a. Celle-ci est en $O(m*p)$, puisque les opérations d'empilement et de dépilement concernent au plus m éléments, avec m est la taille d'une séquence et p est le nombre de séquences analysées. Ainsi le problème d'extraction des sous séquences de longueur inférieure à n est résolu en $O(m*p*log(n))$. Dans le pire des cas, on a $n = m$. Mais en pratique, n est très inférieur à m. La vérification des paramètres α et β, est de complexité $n*m*p$, ce qui fait une complexité globale de $n*log(n)*m^2 * p^2$.

L'approche des n-grammes

Un n-gramme et une sous séquence de n acides aminés (Miller *et al.*, 1999). Dans les travaux antérieurs sur la manipulation de textes, plusieurs algorithmes

d'extraction de motifs ont été développés. On cite l'algorithme naïf, l'algorithme KMP (Knut *et al.*, 1977) et l'algorithme KMR (Karp *et al.*, 1972). Ces algorithmes diffèrent par leurs complexités et par les types de motifs considérés. Dans ce travail, nous avons utilisé l'algorithme naïf pour extraire les n-grammes à partir d'un fichier de protéines.

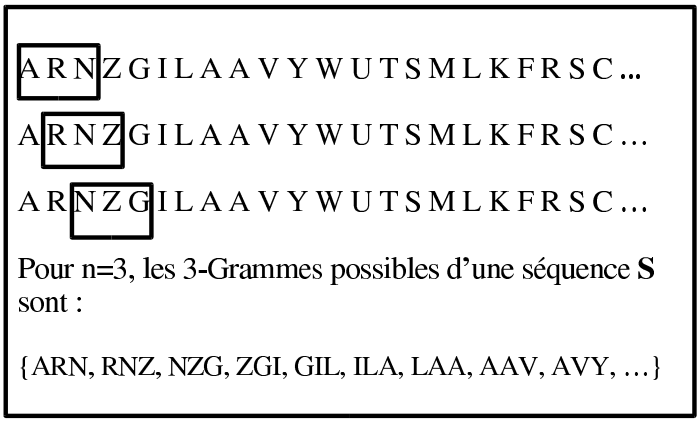


Fig. 4 - Extraction des N-Grammes

Pour une séquence quelconque, l'ensemble des n-grammes pouvant être générés est obtenu en déplaçant une fenêtre de n caractères sur la séquence entière. Ce déplacement s'effectue caractère par caractère. A chaque déplacement la sous séquence des n acides aminés est extraite. La figure 4 illustre ce principe. L'ensemble de ces sous séquences constitue les n-grammes pouvant être générés à partir d'une seule séquence (Miller *et al.*, 1999). Ce processus sera itéré pour toutes les séquences analysées.

La complexité algorithmique est égale à $n*m*p$, avec n la taille d'un n-gramme, m la taille d'une séquence et p le nombre de séquences analysées. Comme il est connu qu'il existe 20 acides aminés, donc, le nombre de n-grammes possibles est égal à 20^n.

Expérimentations et résultats

Pour expérimenter les approches étudiées, on a extrait plusieurs familles de protéines de la banque de données SCOP (Murzin *et al.* 1995). On a essayé de discriminer les familles deux à deux. On s'intéresse, donc, à extraire les n-grammes et les descripteurs discriminants qui discriminent une famille par rapport à une autre. Dans une première étape on a essayé de déterminer la meilleure longueur des n-grammes (la valeur de n), ainsi que les meilleures valeurs des paramètres α et β. Dans la deuxième étape, on a comparé les performances des deux approches étudiées d'extraction de caractéristiques.

1.1 Longueur optimale des n-grammes

Table 1. Recherche de la meilleur taille de *n*

Longueurs des n-grammes	Nombre des n-grammes possibles	Nombre des n-grammes distincts	Filtrer <=05%	Filtrer <=15%	Filtrer <=25%
1	249	20	20	20	20
2	24 962	400	400	396	387
3	59 197	7 145	4 474	1287	464
4	65 367	31 595	2 390	108	15
5	66 125	42 040	1 344	25	4
6	66 189	45 206	1 151	5	1
7	66 135	46 954	1 045	1	0
8	66 136	48 226	984	0	0

Dans un travail ultérieur (Mhamdi *et al.*, 2004), pour calculer la valeur optimale de *n*, on a expérimenté plusieurs valeurs sur un même fichier. Pour déterminer la meilleure taille des n-grammes (*n*), on a éliminé les n-grammes qui ont un nombre d'occurrences <= *x %*, avec $x \in \{5, 15, 25\}$. Le tableau 1 présente les résultats de ce filtrage. Nous remarquons que les n-grammes qui ont une taille $n >= 5$ ont une faible présence. Ces valeurs seront rejetées.

On sait que le nombre des n-grammes possible est égal à 20^n. Pour n=4, on a 20^4=160.000 n-grammes possibles. Ce nombre est trop élevé pour être traité par les techniques de fouille de données. Ce qui nous conduit à choisir la valeur de n=3, avec 20^3=8000 n-grammes possibles. Ces résultats confirment l'étude de Miller (Miller *et al.* 1999).

1.2 Valeurs optimales de α et β

Table 2. Recherche de meilleurs valeurs de α et β

Longueur des DDs		Valeur de α	Valeur de β	Nombre de DDs	Taux d'erreur pour C4.5
7	3	**0**	**0**	17879	Trop de DDs
5	2	**10**	**90**	462	0.1516
5	2	**20**	**80**	604	0.1424
2	3	**30**	**70**	602	0.1290
5	2	**40**	**60**	433	0.0478
3	2	**50**	**50**	241	0.1240

Pour calculer les valeurs optimales de α et β, on a expérimenté plusieurs valeurs sur un même fichier. Pour chaque couple de valeurs de α et β, on a calculé le taux d'erreurs de classification avec le classifieur C4.5 (Lefébure & Venturi 2001). Le tableau 2 présente les résultats de ces expérimentations. Le tableau montre, dans un premier lieu, que les valeurs optimales du couple α et β sont (**α =40, β =60**). Nous remarquons aussi que le nombre de DDs est considérablement réduit par toutes les valeurs non nulles de α et β. Comme troisième résultat, le tableau montre que la taille 3 (qui coïncide avec la taille optimale des n-grammes) est présente dans toutes les expérimentations.

Table 3. Recherche de la taille des DDs les plus fréquents

α, β	**n=2**	**n=3**	**N=4**	**n=5**	**n=6**	**n=7**
0, 0	0	2406	**15147***	315	10	1
10,90	**341***	77	13	1	0	0
20,80	273	**327***	2	2	0	0
30,70	195	**407***	0	0	0	0
40,60	129	**303***	0	1	0	0
50,50	68	**173***	0	0	0	0

Le tableau 3 présente le nombre d'apparitions des différentes longueurs des DDs. Nous remarquons que les DDs de longueur 3 sont, généralement, les plus fréquents.

1.3 Comparaison des deux approches

Pour comparer les deux approches n-grammes et DDs, on les a expérimenté avec 10 fichier de protéines. Premièrement, pour chaque fichier on a extrait tout les n-grammes possibles avec *n*=3 et tous les DDs avec α =0 et β =0. Dans un deuxième lieu on a calculé le taux d'erreurs de classification qui correspond à chaque fichier de n-grammes et de DDs. Les résultats sont présentés dans le tableau 4. Dans l'étape d'extraction des caractéristiques on a calculé le temps moyen d'extraction et de construction du fichier booléen. On a trouvé comme résultat, le temps moyens pour les n-grammes est égal à *48 secondes* tandis que pour les DDs ce temps est égal à *900 secondes*. Nous avons remarqué aussi que le temps d'extraction des DDs augmente considérablement lorsque les paramètres α et β sont non nuls. Ceci est dû au temps de sélection des caractéristiques.

Table 4. Comparaison entre nombres et taux d'erreurs de DDs et 3-Grammes

Fichier de familles	3-grammes		DDs (α =0, β =0)		DDs (α =40, β =60)	
	Nombre	Taux d'erreurs (1-NN)	Nombre	Taux d'erreurs (1-NN)	Nombre	Taux d'erreurs (1-NN)
F_1_2	7145	0.057471	17879	0.1034	433	0*
F_1_3	6998	0.053191	17432	0.0957	417	0*
F_1_4	6945	0.041322	18347	0.0744	453	0*
F_1_5	6828	0.046296	18075	0.0833	447	0*
F_2_3	7143	0.108911	18709	0.1584	242	0.0297
F_2_4	7107	0.062500	19227	0.0297	228	0.0078
F_2_5	7011	0.095652	18704	0.0957	226	0.0087
F_3_4	6860	0.177778	19822	0.2593	245	0.0370
F_3_5	6740	0.262295	19223	0.3115	237	0.0574
F_4_5	6688	0.161074	20149	0.2349	226	0.0201

Le tableau 4[1] montre que le nombre des DDs extraits avec α =0 et β =0, est presque le triple des n-grammes. Ce qui explique que le taux d'erreurs avec les n-grammes est moins élevé qu'avec les DDs. Nous remarquons aussi que les taux d'erreurs de classification pour les DDs extraits avec les valeurs optimales de α et β, identifiées dans le tableau 2 (α =40 et β =60), sont améliorés d'une façon remarquable. Ainsi, le nombre de DDs est spectaculairement diminué. Le taux d'erreurs étant calculé par la méthode Leave-one-out pour le classifieur 1- PPV (Lefébure & Venturi 2001).

Conclusion

Dans ce papier, on a présenté deux approches d'extraction de caractéristiques à partir de séquences protéiques : l'approche des descripteurs discriminants et l'approche des n-grammes. On a réalisé une comparaison théorique et expérimentale entre les deux approches.

D'après l'étude de complexité, nous avons remarqué que l'approche des n-grammes est plus rapide. Ceci est dû au fait que la valeur de *n* (longueur de la sous

séquence) est fixé au préalable. Tantdisque, l'approche des descripteurs discriminants

trouve des sous séquences de longueurs variables. En plus, elle filtre les sous séquences trouvées selon leurs fréquences d'apparition dans les familles à discriminer. Cependant, elle garde l'avantage d'extraire des sous séquences de longueurs variables, tout en garantissant leur minimalité et leur aspect discriminant.

[1] Les valeurs nulles des taux d'erreurs pour les quatre premiers fichier sont expliquées par l'existence d'un DD "LDLS" dans 100% de séquences de la famille F1 et dans 0% de séquences des autres familles.

D'après l'étude expérimentale, nous avons constaté d'abord que la longueur optimale des n-grammes est égale à 3. Ce qui confirme les constatations de Miller (Miller *et al.*, 1999). La même constatation a été confirmée par les expérimentations qu'on a effectuées avec les DDs, où on a remarqué que les sous séquences de longueur 3 sont très fréquentes et présentent dans toutes les expérimentations.

Nous avons constaté aussi que les valeurs optimales des paramètres α et β correspondent à (40, 60). Pour ces mêmes valeurs, le nombre de caractéristiques est relativement réduit par rapport au nombre des n-grammes. Avec ce nombre élevé des n-grammes, il sera difficile d'appliquer les techniques de datamining. D'où la nécessité d'une phase de sélection de variable. Dans cette perspective, on peut envisager l'utilisation des paramètres α et β. Comme on peut envisager le filtrage par le test de Chi-2 (Radwan *et al.*, 2003), et la sélection par les méthodes Wrapper (Isabelle & André, 2003).

Nous avons remarqué aussi que le coût en temps de calcul pour l'extraction des DDs est élevé. Il faut signaler aussi que ce coût augmente spectaculairement lorsqu'on prend des valeurs de α et β non nulles (on impose des contraintes de discrimination). Ceci est dû au fait qu'on sera amené à calculer les fréquences d'apparitions des DDs dans chaque famille.

Il existe d'autres approches d'extraction de variables tels que celle basé sur les modèles de Markov (HMM) (Soumya & Mark, 2001) et les descripteurs de Fourrier (FFT) (Ming *et al.*, 1998), les motifs avec des gaps (Wang *et al.,* 1994), etc. Comme futur travail, on compte étudier ces approches et les comparer avec celles présentées dans ce papier.

Références

Fayyad U. M., Piatetsky-Shapiro G., & Smyth P. (1996). From data mining to knowledge discovery: An overview, in " Advances in Knowledge Discovery and Data Mining", AAAI Press and the MIT Press, chapter 1, pages 1-34.

Maddouri M. & Elloumi M. (2002). A data mining approach based on machine learning techniques to classify biological sequences. Journal of Knowledge Based System, vol. 15. Issue 4. Elsevier Publishing CO. p- 217-223. Amsterdam

Maddouri M. & Elloumi M. (2004). Encoding of primary structures of biological macromolecules within a datamining perspective. Journal of Computer Science and Technology(JCST). Vol. 19. num. 1. Allerton Press. p. 78-88. USA

Hastie T., Tibshirani R. & Friedman J. (2001). The elements of statistical learning. Springer-Verlag.

Lefébure R. & Venturi G. (2001). Data mining : Gestion de la relation client personnalisation de sites Web, Eyrolles.

Karp R. M., Miller R. E. & Rosenberg A. L. (1972). Rapid identification of repeated pattern in strings, Trees and arrays. In Proc 4th ACM Symptom Theory of Computing. p. 125-136.

Miller D., Shen D. Liu J. & Nicholas C. (1999). Performance and scalability of a large-scale N-gram Based Information Retrivial System. Journal of digital information. 1(5).

Knuth D.E,. Morris J.H &. Pratt V.R. (1977). Fast pattern matching in strings. SIAM Journal on computing 6 (1) p. 323-350.

Murzin G. A. , Brenner E. S., Hubbard T. & Chothia C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Bio.. 247,p. 536-540.

Mhamdi F., Elloumi M. & Rakotomalala R. (2004). Text mining, features selection and data mining for proteins classification. In Proc. of 1st International Conference on Informatique & Communication technologies :From Theory to application, Damascus, Syria.

Miller D., Shen D. Liu J. & Nicholas C. (1999). Performance and scalability of a large-scale N-gram Based Information Retrivial System. Journal of degital information. 1(5).

Radwan J., Jérémy C. & Rakotomalala R. (2003). Un cadre pour la catégorisation de textes multilingues. In Proc of 7èmes Journées internationales d'Analyse statistique des Données Textuelles. p 650-660.

Isabelle G. & André E. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research 3: p. 1157-1182.

Soumya R. & Mark C. (2001). Representing sentence structure in Hidden Markov Models for information extraction. In Proc of the 17th Intenational Joint Conference on Artificial Intelligence.

Ming Y. & al. (1998). A new Fourier transform approach for protein coding measure based on the format of the Z curve. Bioinformatics, 14(8), p. 685-690.

Wang L., Chirn W., Marr G., Shapiro A. Shasha D. & Zhang K. (1994). Combinatorial pattern discovery for scientific data: Some preliminary results. In Proc of the ACM SIGMOD International Conference on Management of Data. p. 115-125.

Émergence fonctionnelle d'un micro-organisme par auto-organisation de ses gènes

Carole Bernon, Jean-Pierre Mano, Pierre Glize

Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
118 Route de Narbonne – 31062 Toulouse Cedex 04
{bernon, mano, glize}@irit.fr

Résumé : Cet article présente un premier pas vers l'apprentissage du comportement d'une cellule utilisant une théorie particulière de l'émergence (celle des systèmes multi-agents adaptatifs) en effet, par le calcul, les propriétés fonctionnelles d'une cellule ne peuvent se déduire des propriétés de ses seuls composants. L'organisme « virtuel » élabore par apprentissage un modèle d'interactions entre ses potentialités matérialisées par ses gènes qui doit être cohérent avec les données biologiques expérimentales recueillies, de nature transcriptionnelles, phénotypiques et métaboliques.

1 Introduction

Décrire -et comprendre- le comportement des systèmes vivants dans leur globalité, même lorsqu'il s'agit d'un être unicellulaire, est un objectif de la biologie qui n'est pas encore atteint. La modélisation déductive, qui permet de tester une hypothèse en confrontant ses conséquences aux faits observés, n'a pas abouti. Cette démarche déductive échoue principalement car la fonction d'un élément constitutif du système vivant dépend du fonctionnement simultané des autres éléments. Par le calcul, les propriétés fonctionnelles de la cellule ne peuvent être déduites des propriétés de ses seuls composants. La complexité de la réponse biologique est liée à la transmission de l'information portée par le transcriptome jusqu'au comportement physiologique. Elle sous-entend le métabolome, le fluxome mais également les différents mécanismes de régulation : feedback, régulation génomique, régulation énergétique, homéostasie cellulaire…

Pourtant la génomique fonctionnelle, issue de l'analyse des profils d'expression génique et protéique, est un outil puissant dans l'analyse des systèmes cellulaires par la quantification des variations d'expression des gènes, déterminants de la réaction biologique. La technique des puces à ADN permet d'étudier le transcriptome d'une cellule sous diverses conditions générant ainsi une masse considérable d'informations, concernant l'expression des gènes de cette cellule, qui devront être analysées et interprétées. D'autre part, de nombreuses données macroscopiques (de type biologique et physique) issues d'expérimentations sont disponibles ou réalisables.

Dans le champ de la biologie synthétique, l'objectif du présent projet est de permettre d'établir un modèle informatique simulant le comportement fonctionnel d'un micro-organisme et offrir ainsi des possibilités d'analyse des éléments structurels du monde vivant. Fondé sur une théorie des systèmes complexes, l'« organisme virtuel » élabore par apprentissage un modèle d'interactions entre ses potentialités matérialisées par ses gènes, cohérent avec les réponses expérimentales recueillies (transcriptionnelles, phénotypiques et métaboliques).

La section suivante de cet article présente tout d'abord la technique d'apprentissage adoptée qui repose sur la théorie des systèmes multi-agents adaptatifs. La description du SMA adaptatif solution au problème donné est ensuite décrite en section 3, accompagnée de résultats préliminaires dans la section précédant la conclusion.

2 Apprentissage par émergence d'organisation coopérative

L'apprentissage du comportement d'une cellule à travers des données biologiques expérimentales est un problème complexe (le nombre de gènes dans un micro-organisme est élevé, les données expérimentales à prendre en compte sont nombreuses et hétérogènes), dont l'espace de recherche est gigantesque et irrégulier et qui n'a pas de solution ou d'algorithme de résolution connus. C'est pourquoi nous avons décidé de ne pas employer les techniques d'apprentissage classiques ou des méta-heuristiques [Dréo03] et de l'aborder en employant une théorie particulière de l'émergence : celles des AMAS (Adaptive Multi-Agent Systems).

2.1 Principe des AMAS

Un moyen d'apprendre pour un système qui est plongé dans un environnement changeant est de s'y adapter. L'adaptation peut se réaliser en appliquant le principe d'auto-organisation, principe retrouvé dans certains phénomènes naturels (colonies de fourmis, termites… [Bonabeau97]). Heylighen définit l'auto-organisation comme « l'émergence spontanée d'une cohérence globale à partir d'interactions locales entre des composants initialement indépendants » [Heyligen01]. Depuis quelques années, nous étudions l'auto-organisation comme le moyen de s'affranchir de la complexité et de l'ouverture des applications informatiques à mettre en œuvre [Camps98]. Le résultat de cette étude nous a poussé à proposer une théorie appelée AMAS [Gleizes99] dans laquelle la coopération est le moteur grâce auquel le système s'auto-organise pour apprendre à s'ajuster aux changements de son environnement. Les interactions entre les composants du système dépendent seulement de la vue locale qu'ils possèdent et de leur capacité à coopérer les uns avec les autres. Changer leurs interactions revient à modifier l'organisation du système et à changer ainsi le comportement global de ce système. Ce comportement global émerge des interactions entre composants, il devient inutile de le connaître ou de le décrire, seul le comportement des composants doit être donné.

Dans cette approche des AMAS, le système multi-agent adaptatif est donc composé d'agents dits « coopératifs » dont le cycle de vie consiste à percevoir des choses dans leur environnement, à décider de la manière dont il vont agir en restant le plus coopératif envers les autres, puis à appliquer l'action choisie. Au niveau d'un agent, la coopération est décrite de manière « proscriptive » i.e. qu'au lieu de reconnaître des situations dans lesquelles un agent est coopératif, il sait détecter des situations dans lesquelles il n'est plus coopératif, situations appelées Situations Non Coopératives (SNC). Dès qu'un agent détecte qu'il se trouve dans une telle situation, il agit pour revenir à une situation qu'il juge, de son point de vue, être coopérative envers les autres, mais aussi envers lui-même ; un agent coopératif n'est pas altruiste.

2.2 Contexte expérimental

Le micro-organisme cible de l'étude est la levure « Saccharomyces Cerevisiae » dont le génome, parfaitement connu, comporte 6200 gènes [Alfenore04, Harris01]. Cet organisme a l'avantage d'être un eucaryote unicellulaire possédant des qualités expérimentales (non pathogène, aérobie…) mais aussi un intérêt économique.

Le système informatique devra donc apprendre à fonctionner comme une population de levures en fonction de son environnement, en s'appuyant sur deux types de données expérimentales :

- Des données transcriptomiques obtenues grâce au protocole des puces à ADN puis traitées statistiquement par les chercheurs en biologie[1]. Ces données fournissent notamment une ligne de description de chacun des gènes exprimés durant l'expérimentation effectuée, soit environ 6000 lignes de donnés. Parmi les informations concernant un gène, nous n'en avons retenues que deux pour la réalisation du prototype développé : le nom du gène et son taux de sur- ou sous-expression.

[1] Ce travail est effectué en collaboration avec le Laboratoire Biotechnologies-Bioprocédés (UMR 5504) de l'INSA de Toulouse.

- Des données provenant de la caractérisation macroscopique du comportement de la levure pour des environnements spécifiques de culture. Ces données expriment la quantité des composants macroscopiques (tels que le pH ou le CO2) présents dans le bioréacteur, en fonction du temps ou concernent d'autres facteurs comme la température.

Il est à noter que les deux ensembles de données ne sont pas synchronisés, les prélèvements des deux types de données sont effectués selon des échelle temporelles très différentes (d'un facteur de un à mille), et ne sont pas du même ordre de grandeur. Il a donc été nécessaire d'utiliser une technique d'interpolation temporelle des données en utilisant des splines cubiques.

3 La plate-forme Microméga

L'objectif de notre étude est de construire un système dont les composantes vont apprendre à interagir comme s'il s'agissait d'un système vivant. À partir des données observables, le système artificiel doit parvenir à prédire l'expression génomique de la levure cible en déterminant l'organisation individuelle et collective de ses constituants afin de tendre vers ces données observables. Une fois l'organisme virtuel modélisé, il pourra prédire des données transcriptomiques simulées à partir de nouvelles données macroscopiques, ce qui permettrait de réaliser de nouvelles expériences biologiques simulées sans avoir recours à de réelles puces à ADN, technique onéreuse.

La première étape est de montrer la faisabilité d'un tel simulateur, bâti sur une expression simple des fonctions des gènes, dans leur environnement contraignant, afin d'en faire ressortir une fonction émergente.

3.1.1 Le système multi-agent

Les données transcriptomiques et macroscopiques que le système doit prendre en compte ont besoin d'un représentation dans ce système car elles vont y agir pour ajuster leur quantité en fonction des conditions expérimentales. Afin de modéliser au plus juste le comportement de la levure, d'autres facteurs intermédiaires, intervenant dans la régulation, doivent aussi être pris en compte tels que les protéines ou d'autres substances.

Ces données transcriptomiques, macroscopiques ou de niveau intermédiaire sont vues comme les composantes du système d'apprentissage. Elles sont vues comme des entités autonomes, n'ayant qu'une vue locale de leur environnement, ayant chacune un but propre qui est de satisfaire ses contraintes et de s'ajuster pour exprimer sa bonne quantité dans le système. Ce sont ces données de différentes natures qui vont être représentées par des agents coopératifs (voir figure 1) :

- des agents transcriptomiques chargés de représenter les 6200 instances de gènes de la levure. Ils peuvent avoir des interactions avec tous les autres types d'agents ;
- des agents macroscopiques représentant les 30 instances de composants tels que l'oxygène, l'éthanol… Ils interagissent avec les agents transcriptomiques ou intermédiaires mais n'interagissent pas entre eux car les données correspondantes sont supposées maîtrisées.
- et des agents intermédiaires qui représenteront les autres facteurs, comme les protéines, ils interagissent avec tous les autres types d'agents.

Le SMA est construit comme un réseau dynamique d'agents et de liens. La phase préliminaire de construction de ce réseau consiste à intégrer les agents macroscopiques et transcriptomiques et à leur adjoindre des liens aléatoires. Le nombre de liens, en entrée et en sortie, que peut posséder un agent est fixé arbitrairement. Ces liens correspondent à l'influence d'activation/inhibition qu'un agent peut avoir sur un autre. Ainsi, un lien reliant deux agents se caractérise par :

- Un poids correspondant à une activité de production pour l'agent pour lequel le lien est entrant ;
- Un poids correspondant à une activité de consommation pour l'agent pour lequel le lien est sortant ;

- Un indicateur de confiance qui sert à déconnecter le lien lorsqu'il tombe à 0.

C'est l'attitude sociale d'un agent qui va lui permettre de changer ses liens avec les autres et ainsi de faire émerger une nouvelle organisation du réseau. Le système est ainsi capable de s'adapter et d'apprendre en fonction de ses conditions de fonctionnement.

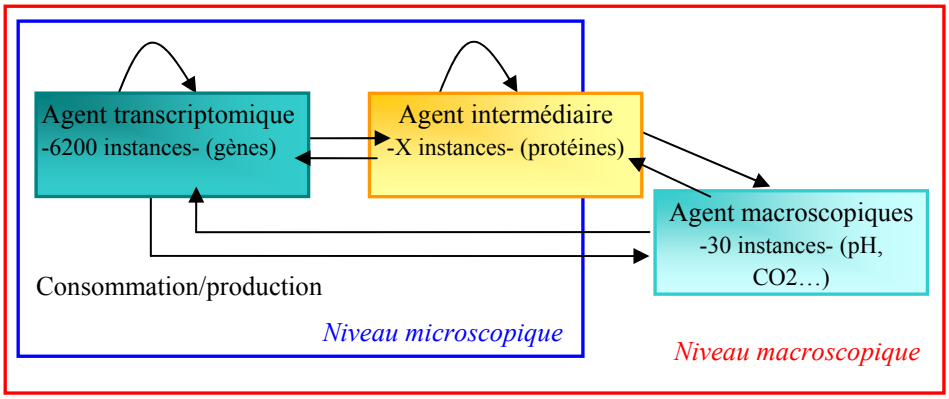


Figure 1. Interactions entre agents dans la plate-forme Microméga.

3.1.2 Comportement coopératif des agents Microméga

Dans les AMAS, la coopération se définit comme suit : (1) tout signal perçu peut être interprété et (2) cette information permet d'aboutir à des conclusions et (3) ces conclusions sont utiles à l'environnement. Dès que l'une de ces conditions n'est plus vérifiée, un agent peut se juger en situation non coopérative. Par exemple, si la condition (1) n'est plus vérifiée, il y a incompréhension, ambiguïté ou incompétence ; ne plus vérifier la condition (2) sera source d'improductivité et des situations d'inutilité, de conflit ou de concurrence peuvent se produire lorsque (3) n'est plus vérifiée. L'agent va alors tenter d'agir, de manière autonome, pour sortir de ces situations.

Ces SNC peuvent être liées au cycle de vie de l'agent, à des perceptions, des décisions ou des actions. Dans le réseau d'agents qui nous intéresse, c'est le cas si un agent trouve qu'il lui manque des liens en entrée (respectivement, en sortie) pour fonctionner. Pour lui, cela signifie être inutile et il informe alors ses entrées (respectivement, sorties) actuelles de son besoin. Il en informe aussi les n entrées (respectivement, sorties) les plus récemment déconnectées. Supposons que deux gènes aient des attributions différentes dans la réalité biologique. Aussi, si l'on rencontre, dans le réseau, deux agents transcriptomiques qui réalisent la même fonction sur un troisième, cette situation est une situation non coopérative de concurrence d'activation pour ce dernier. Il diminue alors le poids de sa plus forte entrée, augmentant l'influence de l'agent correspondant (activation), et diminue le poids de sa plus faible entrée, afin de diminuer l'influence de l'autre agent (inhibition).

Mais, les SNC peuvent aussi être liées à un « feed-back ». Dans cette étude, ce feed-back correspond, pour un agent, à l'écart entre la fonction de référence expérimentale et la fonction réalisée par l'agent. Si un écart significatif existe entre les valeurs données par ces fonctions, on considère cette situation comme étant non coopérative et l'agent doit tout faire pour réajuster sa fonction afin de tendre vers la valeur expérimentale. Ainsi les SNC liées à cet écart seront les suivantes :

- La quantité calculée par un agent, à un instant donné, est inférieure à la quantité référencée par son feed-back. Pour la résoudre, il doit augmenter la quantité calculée en augmentant le poids de ses entrée et diminuant le poids de ses sorties, il peut aussi créer de nouveaux partenariats qui agissent dans le même sens que lui.
- La quantité calculée par un agent, à un instant donné, est supérieure à la quantité référencée par son feed-back. Pour la résoudre, il doit diminuer la quantité calculée en diminuant le poids de ses entrées et augmentant le poids de ses sorties, il peut aussi créer de nouveaux partenariats qui agissent dans le même sens que lui.

Bien entendu, lors des réajustements entrepris par un agent afin d'éliminer une situation non coopérative, de nouvelles SNC peuvent apparaître, pour lui ou d'autres agents, qui, elles aussi, devront être résorbées. De proche en proche, le réseau apprendra à se réorganiser grâce à ces réajustements successifs.

4 Premiers résultats et analyse

Les premières expériences qui ont été menées s'intéressent uniquement à la régulation du réseau dynamique d'agents. Elles doivent permettre de créer initialement un réseau aléatoire d'agents et de visualiser les dépendances entre gènes à travers les liens qui s'établissent entre les agents les représentant. Cette première approche a pour objectif de réaliser uniquement l'apprentissage de l'influence de ces dépendances afin de tendre au mieux vers les données expérimentales.

Grâce à l'interface représentée en figure 2, il est possible de connaître les gènes impliqués dans le réseau d'agents grâce à la liste des identificateurs de gènes donnée dans la fenêtre de gauche (en 1). On peut ensuite en sélectionner un afin de dérouler la liste des agents qui lui sont reliés, en sortie ou en entrée. Il est ainsi possible, après le déroulement de la réorganisation du réseau, de visualiser l'activité temporelle de certains gènes (courbes relatives aux agents YPL279C et YLR340W, visualisées en (2)) ainsi que leurs interdépendances grâce aux variations des liens les reliant et symbolisant l'influence qu'un gène possède sur un autre (les courbes en (3) sont relatives à un lien entre le gène transcrit YLR340W et le gène YPL279C).

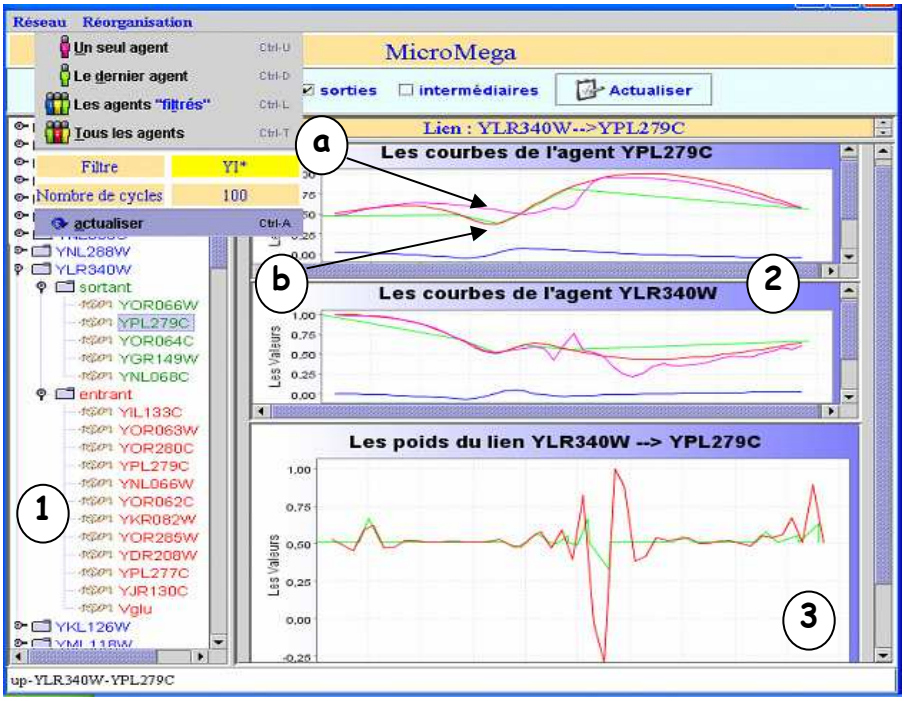


Figure 2. Activité temporelle des gènes transcrits.

Parmi les courbes relatives à un gène, visualisées dans la zone (2), on trouve, notamment, une courbe spline correspondant aux valeurs théoriques expérimentales fournies par les biologistes (b, en rouge) et une courbe spline calculée par l'application (a, en rose). Ces deux courbes permettent d'estimer, pour un agent, l'intégrale de l'écart entre les valeurs expérimentales et les valeurs calculées par l'application (hors figure). On constate qu'au fur et à mesure des cycles de simulation, l'intégrale de cet écart diminue. Néanmoins, elle diminue trop lentement encore. Pour l'instant, il est encore trop tôt pour savoir si cette diminution lente est due à la connectivité aléatoire qui est donnée au début de la simulation mais, il est certain, qu'une amélioration de cette connectivité initiale pourrait améliorer la convergence du résultat.

5 Conclusion et perspectives

Dans le cas qui nous intéresse ici, la difficulté de l'apprentissage réside dans l'espace de recherche gigantesque et chahuté ce qui rendent les techniques informatiques classiques inopérantes. D'autre part, il n'est pas possible de définir une fonction de « fitness » pertinente pour le problème, puisque nous ne connaissons pas le résultat à obtenir, ce qui rend des approches telles que les algorithmes génétiques ou les réseaux de neurones difficilement utilisables.

La résolution par auto-organisation coopérative que nous avons présentée ici permet de diminuer la capacité de calcul nécessaire à la résolution de ce problème en utilisant la coopération en tant qu'heuristique réduisant l'espace de recherche des solutions. Cette approche est d'autant plus intéressante que les règles d'ajustement des influences, les critères de création/suppression de liens et de création/suppression d'agents intermédiaires sont uniquement dictées par des critères locaux fondés sur les situations de non coopération. Et même si beaucoup de travail reste à faire avant de simuler totalement le comportement d'une cellule, la plate-forme Microméga montre que l'apprentissage du comportement des gènes d'une cellule grâce à cette approche est parfaitement viable. Les étapes ultérieures de développement devront implanter des règles de réajustement autonome des liens entre agents (création/suppression) afin de faire évoluer dynamiquement le réseau et de lui permettre de tendre vers un état « idéal ». De plus, il serait utile de pouvoir ajouter des agents intermédiaires, toujours de manière autonome, afin, par exemple, d'étudier l'influence qu'ils peuvent avoir sur les autres agents.

Bien évidemment, certaines approches existantes visent à décrire le comportement d'une cellule. « L'E-Cell », par exemple, quantifie l'activité cellulaire à partir de cinétiques parfaitement caractérisées expérimentalement sur le plan catalytique et quantitatif [Takahashi04] et la « cellule virtuelle » regroupe un ensemble de logiciels permettant de décrire l'organisation métabolique d'un organisme, d'estimer la distribution de la matière dans son fonctionnement et d'extrapoler ses potentialités de production [Loew01]. Mais ces approches reposent sur des modèles partiels. Un défi serait de rechercher une description précise d'une cellule vivante basée sur des mécanismes et des valeurs paramétriques expérimentalement déterminées. L'approche que nous avons présentée ici est un premier pas dans cette voie.

Remerciements – Nous tenons à remercier Jean-Louis Uribellarea, professeur au LBB de l'INSA, et Elsa Macchion pour le travail effectué sur cette première version de la plate-forme Microméga durant son stage d'ingéniorat CNAM.

Bibliographie

[Alfenore04] Alfenore S., Cameleyre X., Benbadis L., Bideaux C., Uribelarrea J.-L., Goma G., Molina-Jouve C., et Guillouet S.E., Aeration Strategy: a Need for Very High Ethanol Performance in Saccharomyces cerevisiae Fed-batch Process, Applied Microbiology and Biotechnology .65: 537-542, 2004.

[Bonabeau97] Bonabeau E., Theraulaz G., Auto-organisation et comportements collectifs : la modélisation des insectes sociaux, Auto-organisation et comportement, Ed. Hermès, 1997.

[Camps98] Camps V., Gleizes M_P., et Glize P., Une théorie des phénomènes globaux fondée sur des interactions locales - Sixièmes journées francophones IAD&SMA, Pont-à-Mousson, Ed. Hermès, Nov. 1998.

[Dréo03] Dréo J., Pétrowski A., Siarry P. et Taillard E., Méta-heuristiques pour l'optimisation difficile – Ed. Eyrolles, 2003.

[Gleizes99] Gleizes M-P., Camps V., et Glize P., A Theory of Emergent Computation Based on Cooperative Self-Oganization for Adaptive Artificial Systems, Fourth European Congress of Systems Science, Valencia, 1999.

[Harris01] Harris M.A., Issel-Tarver L., Schroeder M., Botstein D. et Cherry J.M., Saccharomyces Genome Database Provides Tools to Survey Gene Expression and Functional Analysis Data, Nucleic Acids Res, 29, 80-81, 2001.

[Heylighen01] Heylighen F., The Science of Self-organization and Adaptivity, In The Encyclopedia of Life Support Systems, (EOLSS Publishers Co. Ltd), 2001.

[Loew01] Loew L.M., Schaff J.C., The Virtual Cell: a Software Environment for Computational Cell Biology, Trends-in-biotechnology-Regular-ed., 19(10):401-406, 2001.

[Takahashi04] Takahashi K., Kaizu K., Hu B. et Tomita M., A Multi-algorithm, Multi-timescale Method for Cell Simulation, Bioinformatics, 20(4), 538-546, 2004.

Automatic extraction of relevant nodes in biochemical networks

Sébastien Vast, Pierre Dupont, Yves Deville

Department of Computing Science and Engineering
Université catholique de Louvain
Place Sainte Barbe, 2
B-1348 Louvain-la-Neuve - Belgium
{svast, pdupont, deville}@info.ucl.ac.be
http://www.info.ucl.ac.be/~{svast,pdupont,yde}

Abstract : In this paper we describe a novel method for extracting a set of nodes that best capture the connections between k given nodes of interest in a biochemical network. This method relies on the projection of the nodes of the network, seen as an undirected graph, into an euclidean space. Euclidean distances between nodes in the projected space correspond to their commute time distances in the original graph, a measure based on a random walk model on the graph. Commute time reflects the distance between two nodes while considering all paths connecting them. Results on artificial data illustrate the interest of this approach.

Keywords: biochemical network analysis, subgraph mining, commute time distance, spectral graph analysis

1 Introduction

Biochemical networks model interactions between biochemical entities within cells. Metabolism can be viewed as a network of chemical reactions catalyzed by enzymes, and connected via their substrates and products; a metabolic pathway is then a coordinated series of reactions. Other types of biochemical networks include regulatory or signal transduction networks. Several models exist to represent biochemical networks (Deville *et al.*, 2003). In most cases, these networks can be viewed as directed or undirected graphs. The present work is part of the BioMaze project which aims to produce computer tools for analyzing biochemical networks. BioMaze extends the Amaze project which aims to build a biochemical database integrating the three types of networks mentioned above and to provide dedicated query tools (van Helden *et al.*, 2000).

The specific problem we address here is the extraction of a relevant subgraph of an undirected[1] graph, which best explains the relations between k given nodes of interest in this graph. Assume, for instance, we are analyzing the synthesis of *pyruvate* from *glucose* and would like to study the possible influence of the expression of a given *gene* on a protein, say *phospho-fructokinase-2*, in the context of the regulation of this metabolic pathway. In this case, we have 4 nodes of interest in a possibly very large graph of interactions and we would like to extract a relevant subgraph explaining the relations between these 4 nodes. The methods described in this paper are also applicable to other practical domains.

This paper presents a novel approach to this problem. It relies on the projection of the nodes of the graph into an euclidean space. Euclidean distances between nodes in the projected space correspond to their *commute time distances* in the original graph, a measure based on a random walk model on the graph (Saerens *et al.*, 2004). Commute time reflects the distance between two nodes while considering all paths connecting them. This contrasts with simpler approaches which would extract only specific paths between each pair of nodes of interest, such as shortest distance or maximal flow paths. Here the goal is the extraction of a relevant subgraph as this is considered to be more informative. An inspiring approach to this problem was presented recently in (C. Faloutsos & Tomkins, 2004) but the problem was restricted to 2 nodes of interest. We adopt here a different point of view allowing for a direct solution to the general problem with any number of nodes of interest. We propose to solve the problem in two steps: the extraction of a subset of relevant nodes in the graph followed by the construction of a subgraph connecting them. The present contribution focuses on the first step.

Section 2 proposes a formal statement of the problem we address. Some possible methods to solve it are discussed and contrasted with our approach. The theory behind the notion of commute time distance is summarized in section 3. Section 4 details how to use commute time distances in order to extract a subset of relevant nodes in a graph. Practical experiments are presented in section 5.

2 The problem of extracting a subset of relevant nodes

Problem statement: **Given** a connected undirected graph $G = (V, E)$, where V denotes a set of nodes (or vertices) and E denotes a set of weighted edges, a non-empty set $K \subseteq V$ of nodes of interest and s a strictly positive integer, **find** a set $S \subseteq V \setminus K$ of nodes, with $|S| = s$, optimizing a goodness function $g(S, K)$. The goodness function $g(S, K)$ measures how well the s *extracted nodes* explain the relations between the $k = |K| \geq 2$ *nodes of interest* in the graph.

The goodness function should measure how well the nodes of interest are connected through paths to which the extracted nodes belong. A naive approach to this problem consists in extracting nodes belonging to shortest paths between pairs of nodes of interest. Consider, for instance, the graph depicted in Figure 1 and assume this graph represents a road map between cities A and B (*i.e.* $k = 2$, in the present case). The

[1] Even though there is a direction of flow in a metabolic pathway, the type of graph analysis considered here does not require directed edges.

shortest distance[2] approach would typically select nodes C and D belonging to the highway connecting A and B. However, as soon as one edge is removed along this path (e.g. in case of a traffic jam) no alternative route from A to B goes through C or D. Nodes included in the dashed circle are more relevant here as they belong to many alternative routes connecting A and B, even though none of these routes might be shorter than the highway. Thus the goodness function should take into account many alternatives routes, possibly all of them.

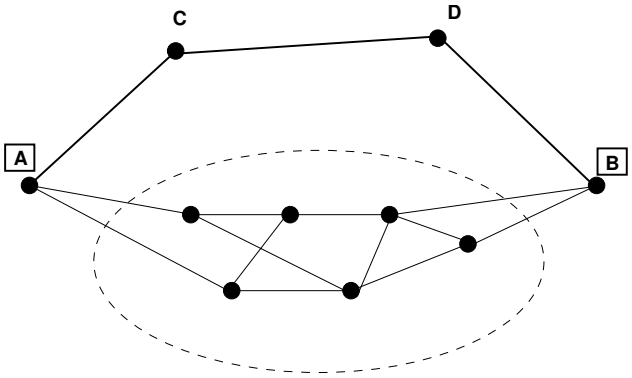


Figure 1: Nodes that best capture the connections between A and B are in the dashed circle as they belong to many alternative routes from A to B, or conversely.

Faloutsos et al. proposed an interesting approach to the more general problem of extracting a relevant subgraph (C. Faloutsos & Tomkins, 2004). This approach can directly be applied to our problem of extracting a subset of the graph nodes. They restrict their attention to the case for which $k = 2$. The goodness function $g(S, K)$ is based on an electrical analogy. The 2 nodes of interest are respectively considered to be the source and the sink of an electrical current. The algorithm searches the paths followed by the current flow and maximizes the sum of current flow in the extracted subgraph. In addition, each node includes some current loss in order to penalize long paths and very highly connected nodes (hubs)[3]. This approach takes into account several paths between the nodes of interest. The fraction of current captured by the subgraph depends on the number and weight of such paths. One drawback of this method is that, due to the current loss, the solution depends on which of the 2 nodes of interest is chosen to be the current source. We propose in the present work an alternative method which deals with any number of nodes of interest with no preference a priori defined between them.

[2]This distance may correspond to the travel time in this particular case.

[3]While it is interesting to extract nodes offering alternative routes to the nodes of interest, hubs do not explain well the specific relations between the nodes of interest as they are well connected to most nodes.

3 Euclidean commute time distance

As motivated by the discussion in section 2, we are looking for a measure describing how well several nodes are connected in a graph by considering all possible paths connecting them. This measure will then be applied to the extraction of a subset of relevant nodes in a graph as detailed in section 4.

The proposed measure relies on a random walk model on the graph. This model assigns transition probabilities to the edges, so that a random walker will jump from one node to another with a probability proportional to the weight of the edge connecting them. The *average commute time*[4] between nodes i and j computes the average time taken by a random walker for reaching node j from node i, and coming back to i. The square root of this quantity is a distance measure between any two nodes called the *euclidean commute time distance* (ECTD). Most of the theory, summarized in the present section, was introduced in (Saerens *et al.*, 2004). The application of this distance measure to the extraction of a subset of relevant nodes in a graph is detailed in section 4.

Section 3.1 introduces some notations and, in particular, the Laplacian matrix $\mathbf{L}$ of a graph. Section 3.2 details how to compute the ECTD from $\mathbf{L}$.

3.1 The Laplacian matrix of a weighted graph

We consider a weighted undirected graph $G = (V, E)$ with strictly positive weights between each pair of connected nodes. The graph order $|V|$ is also denoted n in the sequel. The larger the weight w_{ij} of the edge connecting node i to node j, the easier the communication between i and j is assumed to be. Moreover, the weights are required to be symmetric ($w_{ij} = w_{ji}$). The *adjacency matrix* $\mathbf{A}$ is defined in the usual way:

$$a_{ij} = \begin{cases} w_{ij} & \text{, if node } i \text{ is connected to node } j \\ 0 & \text{, otherwise.} \end{cases}$$

The diagonal *degree matrix* $\mathbf{D}$ is defined as follows. $d_{ii} = \sum_{l=1}^{n} a_{il}$ and $d_{ij} = 0$, if $i \neq j$. A related quantity is the *graph volume*, that is the sum of node degrees: $D_G = \sum_{i=1}^{n} d_{ii}$.

The *Laplacian matrix* $\mathbf{L}$ of the graph is defined as $\mathbf{L} = \mathbf{A} - \mathbf{D}$. When G has a single connected component, the rank of $\mathbf{L}$ is $n - 1$. Moreover, one can easily show that L is symmetric and positive semidefinite (Chung, 1997).

3.2 Computation of the commute time distances

Klein and Randic proposed in (Klein & Randic, 1981) a distance measure between graph nodes, called *resistance distance* which has the property of decreasing when the number of paths between two nodes increase. As shown by Chandra (Chandra *et al.*,

[4]This notion of commute *time* is equivalent to the average *number of steps* a random walker would make on average to commute between both nodes, since the random walker is assumed to make one step at each time clock.

1989), this measure can be expressed in terms of the random walk model described below.

A random walk on a graph is a Markov chain describing the sequence of nodes visited by a random walker. A state of the Markov chain is associated with every node of the graph. A random variable $X(t)$ represents the current state of the Markov chain at time t. The probability of transiting to state j at time $t+1$, given the current state is i at time t, is given by:

$$P(X(t+1) = j | X(t) = i) = p_{ij} = a_{ij}/d_{ii}.$$

Thus, from any state i, the probability to jump to a state j is proportional to the weight a_{ij} of the edge between i and j. The transition matrix $\mathbf{P} = [p_{ij}]$ of the Markov chain is related to the degree and adjacency matrices as $\mathbf{P} = \mathbf{D^{-1}A}$.

The *average first-passage time* $m(j|i)$ is defined as the average number of steps a random walker, starting in state i, will take to reach state j for the first time. These measure can be computed by the following recurrence (Norris, 1997):

$$\begin{cases} m(j|i) = 1 + \sum_{l=1, l\neq j}^{n} p_{i,l}\, m(j|l) \text{ for } i \neq j \\ m(j|j) = 0 \end{cases} \tag{1}$$

A closely related measure is the *average commute time*, $q(i,j)$, defined as the average number of steps a random walker, starting in state i, will take to enter state j for the first time, and go back to state i for the first time: $q(i,j) = m(j|i) + m(i|j)$. Note that, in general, $m(i|j) \neq m(j|i)$, while the average commute time is symmetric by definition. As shown by several authors, the average commute time is a distance (Klein & Randic, 1981; Gobel & Jagers, 1974). Moreover the square root of the average commute time defines an euclidean distance (Saerens *et al.*, 2004).

A first method for computing euclidean commute time distances is based on the iterative solving of the recurrences (1). An alternative approach derives from the Moore-Penrose pseudoinverse of the Laplacian $\mathbf{L}$, denoted by $\mathbf{L}^+$, as proposed in (Saerens *et al.*, 2004):

$$q(i,j) = D_G(l^+_{ii} + l^+_{jj} - 2l^+_{ij}) \tag{2}$$

If we further define $\mathbf{e_i}$ as the ith column of the $n \times n$ identity matrix, equation (2) can be rewritten as

$$q(i,j) = D_G(\mathbf{e_i} - \mathbf{e_j})^{\mathbf{T}}\mathbf{L}^+(\mathbf{e_i} - \mathbf{e_j}), \tag{3}$$

where each node i is represented by a unit base vector $\mathbf{e_i}$. These nodes can be mapped into an euclidean space that preserves the commute time distances as $\mathbf{L}^+$ is positive semidefinite. Indeed, every positive semidefinite matrix can be transformed to a diagonal matrix (see, e.g., (Meyer, 2000)), $\mathbf{\Lambda} = \mathbf{U}^{\mathrm{T}}\mathbf{L}^+\mathbf{U}$, where $\mathbf{U}$ is an orthonormal matrix made of the eigenvectors of $\mathbf{L}^+$, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n-1}, \mathbf{u}_n = \mathbf{0}]$. Hence, the commute time distances can be rewritten as:

$$q(i,j) = D_G\,(\mathbf{x}'_i - \mathbf{x}'_j)^{\mathrm{T}}(\mathbf{x}'_i - \mathbf{x}'_j) \tag{4}$$

where the following transformations have been applied: $\mathbf{x}_i = \mathbf{U}^{\mathrm{T}}\mathbf{e}_i$, and $\mathbf{x}'_i = \mathbf{\Lambda}^{1/2}\mathbf{x}_i$.

So, in this n-dimensional Euclidean space, the transformed node vectors, $\mathbf{x}'_i$, are exactly separated by euclidean commute time distances (up to the scaling factor D_G).

Close points in this *ECTD space* represent nodes well connected in the original graph G according to any possible paths between them, and the euclidean distance between them measures this connectivity in the original graph.

The ECTD space has dimensionality n, the graph order, but projection to a subspace preserving as much information as possible can reduce computation time. The so-called spectral (or eigenvector) decomposition of $\mathbf{L}^+$ is given by:

$$\mathbf{L}^+ = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}} = \sum_{l=1}^{n-1} \lambda_l \, \mathbf{u}_l \mathbf{u}_l^{\mathrm{T}} \tag{5}$$

where $\lambda_1 > \lambda_2 > \ldots > \lambda_{n-1} > \lambda_n = 0$ are the eigenvalues of $\mathbf{L}^+$, and $\mathbf{u}_l$ the associated eigenvectors. The eigenvector expansion of $\mathbf{L}^+$ can be computed up to $m < n-1$, by considering only the m largest eigenvalues of $\mathbf{L}^+$. This gives rise to an m-dimensional subspace where the commute time distances are approximately preserved.

Finally, since $\mathbf{L}$ and $\mathbf{L}^+$ have the same set of eigenvectors but inverse (non zero) eigenvalues, we do not need to explicitly compute the pseudoinverse of $\mathbf{L}$. It is only necessary to compute the smallest non zero eigenvalues of $\mathbf{L}$, which correspond to the largest eigenvalues of $\mathbf{L}^+$, and their associated eigenvectors. Fast iterative methods exists for this purpose (Golub & Loan, 1996; Sorensen, 1996). The complexity for computing one eigenvalue/eigenvector is $O(n^2)$ and the overall complexity for this method is thus $O(mn^2)$.

4 Node subset minimizing euclidean commute time

The relevant node subset problem can be easily formulated and solved using the euclidean commute time distances between any graph nodes and the nodes of interest. More specifically, we consider the following goodness function :

$$g(S, K) = \sum_{i \in S} d_r(i, K) \tag{6}$$

with

$$d_r(i, K) = \min_{W \subseteq K, |W| = r} \sum_{j \in W} q(i, j)$$

Thus, the contribution of each extracted node to the goodness of the subset S is the sum of the commute time distances to its r ($1 \leq r \leq k$) closest nodes of interest in the ECTD (sub-)space. In the experiments reported below, we considered the distances to the two closest nodes of interest, for each extracted node ($r = 2$). The choice $r = k$ would correspond to considering the distances to all nodes of interest. On one hand, this would allow to take into account the connectivity to all nodes of interest. On the other hand, as this measure would be more global, the extracted nodes might not be particularly well connected to any specific node of interest. We will further study this trade-off in our future work.

Computing an optimal S, which minimizes g for a given number s of nodes to be extracted, is straightforward once the commute time distances between any node of

interest and the other nodes of the graph have been computed. It simply amounts to compute $d_r(i, K)$ for each possible node i of the graph (except the k nodes of interest themselves) and to return the nodes with the s smallest values.

Figure 2 presents the graph of Figure 1 with nodes indexed in increasing order according to $d_2(i, K)$ (here, $K = \{A, B\}$). Unit weight edges were considered in this example.

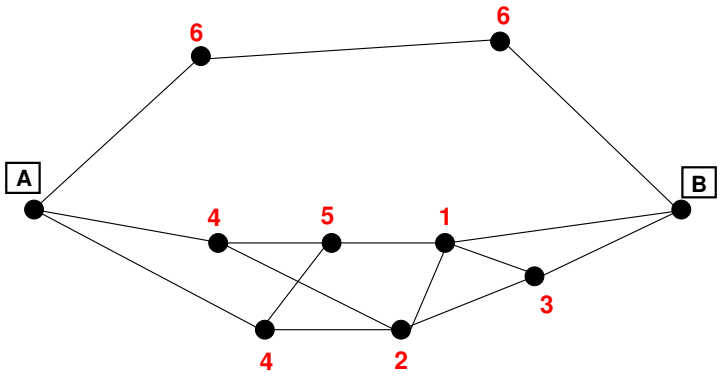


Figure 2: The nodes of this graph are labeled in increasing order (ties are assigned the same rank) according to the sum of their commute time distances to A and B respectively.

5 Experiments

The ultimate objective of this work is to provide a method for a biologist to automatically extract a subset of relevant nodes related to given nodes of interest in a large biochemical network. In order to assess the performance of the proposed method, preliminary experiments with artificial graphs are reported here.

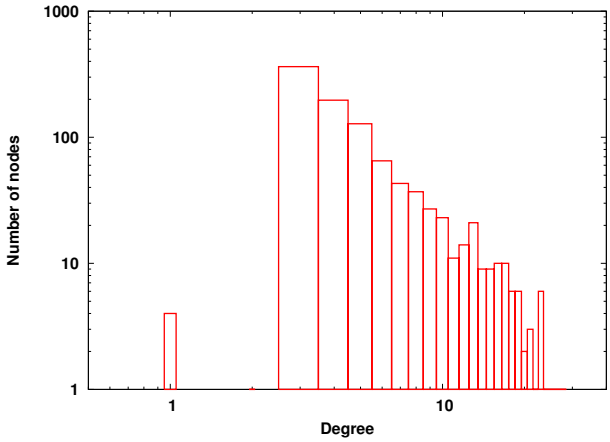


Figure 3: Average degree distribution of graphs used for testing.

A set of 10 graphs of 100 nodes were randomly generated using a power-law graph generator (Barabasi *et al.*, 2000). For each graph, five nodes were used as initial seeds. Next, 95 nodes were iteratively added and randomly connected to 3 nodes of the current

graph. At each step, the probability of an existing node to be connected to the new node is proportional to its current degree. Each generated graph contains a single connected component and all edges have a unit weight. The degree distribution (averaged over all generated graphs) is depicted (in log scales) in Figure 3.

For each graph tested, 10 sets of k nodes of interest were randomly selected. Results are reported for $k = 2, 4$ and 8. In each case, an increasing number of s nodes were extracted. The distance measure $D = \frac{\sum_{i \in S} d_2(i,K)}{\sum_{i \in V \setminus K} d_2(i,K)}$ is the cumulated distance of the subset S of extracted nodes relative to the distance of the total set $V \setminus K$ of nodes which can possibly be extracted. As we aim at minimizing a distance in this case, the smaller D the better.

Comparative results with the method proposed by Faloutsos et al. (C. Faloutsos & Tomkins, 2004) are possible when $k = 2$. These results are presented in Figure 4. Both approaches perform very similarly in this setting, showing that they capture essentially the same information (at least for the tested graphs). However, Faloutsos method cannot extract more than 29 % of nodes in this case (28 out of the 98 nodes which can potentially be extracted with our approach). This comes from the fact that this method only extracts nodes on loopless paths between the 2 nodes of interest. Hence a significant fraction of the graph nodes (here 71 %) may not respect this constraint. One one hand, this illustrates an advantage of our approach. On the other hand, Faloutsos method is more general as it does not only extract a node subset but a connected subgraph. Extension of our method to deal with this more general problem is part of our future work.

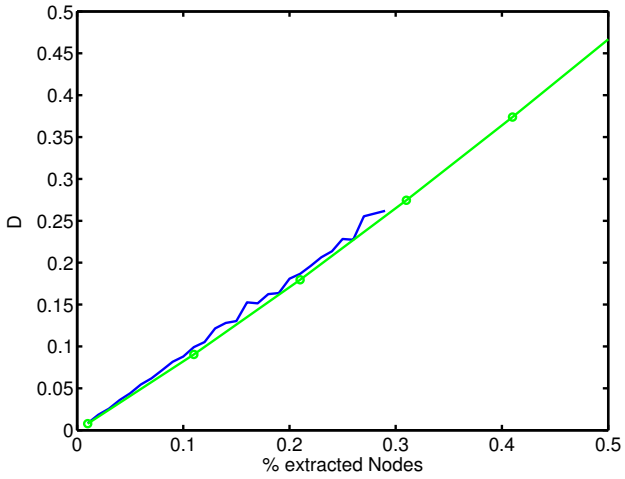
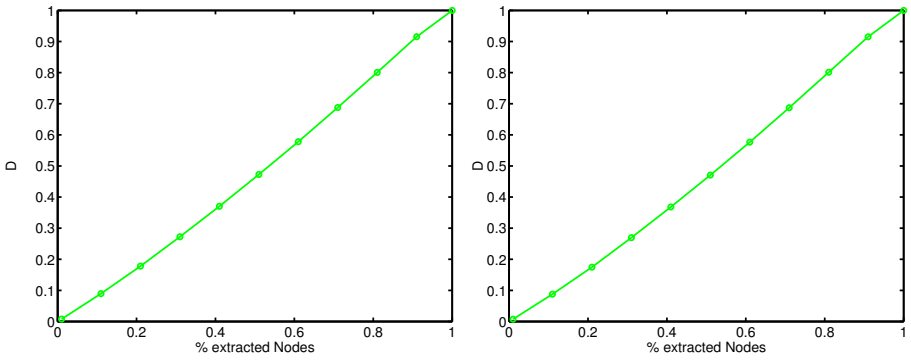


Figure 4: Distance of the extracted subsets for increasing number of extracted nodes. The x axis gives the value of $\frac{s}{n-k}$, that is the percentage of extracted nodes. The green curve (circles) corresponds to our approach minimizing commute times, while the blue curve corresponds to the method of Faloutsos. Results are obtained for $k = 2$ and averaged over 100 tests.

Figure 5 illustrates the results of our approach for $k = 4$ and 8 on the same graphs. Both curves behave similarly and illustrate the generalization of our approach to larger sets of nodes of interest. Note that the computational complexity remains essentially the same as it is dominated by the computation of the same commute time distances in all cases.



(a) $k = 4$ (b) $k = 8$

Figure 5: Distance of the extracted subsets for increasing number of extracted nodes for $k = 4$ or $k = 8$.

In all results presented so far, all edge weights were assumed to be equal (standard deviation $\sigma = 0$). Figure 6 presents the extraction results for another set of 10 graphs of 200 nodes for which a normal distribution on edge weight with a much larger standard deviation ($\sigma > 200$) was defined. As slightly better performance is obtained for these graphs as the distribution of commute time distances is sharper in this case.

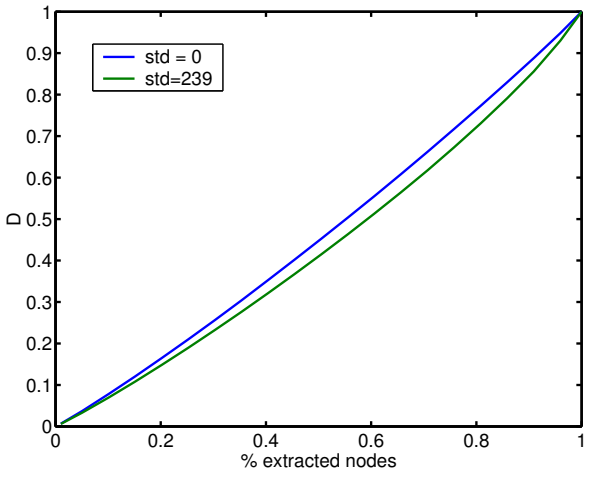


Figure 6: Extraction results for equal weight for all edges ($\sigma = 0$) or large standard deviation ($\sigma = 239$) for a normal weight distribution. Average results for 100 tests with $k = 2$ (10 randomly selected pairs of nodes of interest for each graph).

6 Conclusion and future work

We propose in this paper a novel approach to the extraction of nodes in a biochemical network which best explain the connections between k given nodes of interest in this network. This approach uses commute time distances between nodes, a measure of how well two nodes are connected in a graph by considering all possible paths between them. It is based on the projection of the nodes of the network, seen as an undirected graph, into an euclidean space. Euclidean distances between nodes in the projected ECTD space correspond to their commute time distances in the original graph.

Several questions need to be addressed in the future.

1. The commute time distances can be approximated if the nodes of the original graph are projected into a subspace of the full ECTD space. A lower dimension subspace corresponds to a coarser approximation to the actual commute times while reducing the computational complexity. We will study the trade-off between this complexity and the quality of the set of extracted nodes.

2. Our goodness measure for the extracted node subset is based on the commute time distance from each extracted node to its two closest nodes of interest. As discussed in section 4, alternative goodness measures will be investigated.

3. The more general problem of a relevant subgraph extraction will be considered. Starting from the set of extracted nodes, some edge selection in the original graph has to be designed. This should be derived from the fraction of edges responsible for the largest part of the commute time distances between nodes.

4. Actual experiments on real biochemical networks and result interpretations by biologists are also part of the current project. Comparisons between extracted subgraphs and known pathways could be performed in this regard.

References

BARABASI A., ALBERT R., JEONG H. & BIANCONI G. (2000). Power-law distribution of the world wide web. *Science*, **287**(12).

C. FALOUTSOS K. M. & TOMKINS A. (2004). Fast discovery of connection subgraphs. In *10th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, volume 2, p. 118–127.

CHANDRA A. K., RAGHAVAN P., RUZZO W. L. & SMOLENSKY R. (1989). The electrical resistance of a graph captures its commute and cover times. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, p. 574–586: ACM Press.

CHUNG F. (1997). *Spectral graph theory*. American Mathematical Society.

DEVILLE Y., GILBERT D., VAN HELDEN J. & WODAK S. J. (2003). An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, **4**(3), 246–259.

GOBEL F. & JAGERS A. (1974). Random walks on graphs. *Stochastic Processes and their Applications*, **2**, 311–336.

GOLUB G. H. & LOAN C. F. V. (1996). *Matrix Computations, 3rd edition*. Johns Hopkins University Press.

KLEIN D. & RANDIC M. (1981). Resistance distance. *Journal of Mathematical Chemistry*, **12**, 81–95.

MEYER C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics.

NORRIS J. (1997). *Markov Chains*. Cambridge University Press.

SAERENS M., FOUSS F., YEN L. & DUPONT P. (2004). The princial components analysis of a graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, volume 3201 of *Lecture Notes in Artificial Intelligence*, p. 371–383: Springer-Verlag.

SORENSEN D. C. (1996). *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*. Rapport interne TR-96-40, Rice University.

VAN HELDEN J., NAIM A., MANCUSO R., ELDRIDGE M., WERNISCH L., GILBERT D. & WODAK S. (2000). Representing and analyzing molecular and cellular function using the computer. *Biol. Chem.*, **381(9-10)**, 921–935.

Conformation de biomolécules et apprentissage des interactions de van der Waals par un système multi-agent adaptatif

Camille Besse, Carole Bernon

Institut de Recherche en Informatique de Toulouse,
Université Paul Sabatier,
118, route de Narbonne, 31062 Toulouse Cedex 4
{besse,bernon}@irit.fr

Problématique

Les conformations de biomolécules expliquent leurs rôles structuraux et/ou fonctionnels dans la nature. Seulement, la complexité et la méconnaissance des influences inter-atomiques ne permettent pas de calculer ces conformations de manière efficace. Des approches basées sur des méthodes d'optimisation globale (Fan, 2002) existent mais se trouvent limitées par leurs hypothèses sur l'évaluation de la fonction énergétique à minimiser. L'approche proposée ici, a comme objectif d'apprendre ces interactions atomiques puis de les composer, de manière simple, locale et efficace, en utilisant une technologie basée sur les systèmes multi-agents adaptatifs. Dans un premier temps, elle a pour but de résoudre des problèmes de conformation par minimisation de l'énergie de Lennard-Jones induite des interactions de van der Waals, puis, dans un deuxième temps, elle sera utilisée pour apprendre ces interactions.

La résolution émergente par auto-organisation coopérative

Un moyen d'apprendre pour un système S consiste à transformer sa fonction actuelle f_S de manière autonome afin de s'adapter à l'environnement, considéré comme une contrainte qui lui est donnée. Chaque partie P_i d'un système S réalise une fonction partielle f_{P_i} de la fonction globale f_S. Elle est le résultat de la combinaison des fonctions partielles f_{P_i}. La combinaison étant déterminée par l'organisation courante des parties, il s'ensuit que transformer l'organisation conduit à changer la combinaison des fonctions partielles f_{P_i} et donc à modifier la fonction globale f_S, devenant par là même un moyen d'adapter le système à l'environnement. La justification théorique qui guide le processus d'auto-organisation dans les AMAS est le fait qu'un système qui possède ses parties en interactions coopératives permanentes, réalise la fonction souhaitée dans son environnement (Georgé *et al.*, 2003). En elle-même, l'organisation qui émerge est une organisation observable non prédonnée par le concepteur du système. Mais le plus interessant, c'est l'émergence de la fonction du système qui est produite par l'organisation entre les agents à un instant donné. Pour réaliser cela, tout agent est programmé pour chercher à être en situation coopérative avec les autres agents du système : idéalement, il devrait recevoir des informations pertinentes pour réaliser sa fonction et transmettre ses résultats vers ceux qui devraient en tirer bénéfice. L'agent réalise en permanence sa fonction partielle, mais il agit simultanément sur l'organisation interne du système s'il détecte des situations non coopératives (SNC). Ainsi, la recomposition des fonctions partielles réalisées par chaque agent amène une transformation de la fonction globale du système et les états non coopératifs dus aux situations imprévues sont progressivement supprimés. La résolution de problèmes avec les AMAS s'articule donc autour de trois notions essentielles : apprentissage, auto-organisation et émergence. Il y a apprentissage car le processus de résolution est incomplètement spécifié : au cours de son activité le système doit progresser à partir d'observations sur l'environnement. Cet apprentissage se réalise par un processus d'auto-organisation (coopérative pour les AMAS) entre ses parties constituantes (que nous nommons agents). La solution obtenue est émergente car, ni la spécification initiale du système ni son algorithme d'apprentissage n'ont à connaître ce qui devra être appris.

La résolution émergente de la conformation de molécules

Cette approche ayant fourni des résultats satisfaisants dans des domaines aussi variés que complexes comme la prévision de crues ou encore la résolution d'emploi du temps, nous l'avons appliqué à la recherche de conformations de protéines avec pour objectif à terme de simuler du "docking" de molécules, domaine important en bioinformatique.
Une molécule est un AMAS constitué d'agents coopératifs (les atomes), sa description spatiale est fournie en entrée à l'application suivant un fichier au format de la Protein Data Bank (PDB, 2003). Chaque atome de cette molécule est constamment en interaction avec ses voisins. Nous avons choisi de modéliser ces interactions via l'énergie potentielle de Lennard-Jones exprimée comme suit :

$$E_{vdW}(eV) = \varepsilon \left[\left(\frac{R_0}{d} \right)^{12} - 2 \left(\frac{R_0}{d} \right)^{6} \right] + \frac{A_3}{d^3} + \frac{A_1}{d} \quad (1)$$

où R_0 représente la distance à laquelle l'énergie potentielle est minimale, ε, $A_3 et A_1$ permettent de régler la valeur de ce minimum et les coefficients de la pente de la courbe et d la distance entre deux atomes. $R_0, \varepsilon, A_3, A_1$ sont les paramètres que nous nous proposons d'apprendre.
Cette modélisation nous a permis d'utiliser les interactions atomiques comme moteur du repliement vers la conformation de minimum énergetique (Besse, 2003). Nous avons donc cherché à minimiser cette énergie sans connaitre les paramètres de l'énergie de Lennard-Jones. Cette minimisation se fait de manière coopérative : chaque atome tend à diminuer l'énergie potentiellle résultant des interactions de Van der Waals, sans jamais accroitre la situation énergétique de son voisin de plus grande énergie. De cette manière, l'énergie converge par étapes locales vers un minimum global.
L'apprentissage des paramètres de la loi de Lennard-Jones peut donc être mis en place en mesurant l'écart entre la conformation souhaitée (donnée par une molécule connue) et la conformation obtenue après utilisation de notre algorithme sur cette même molécule déformée par un placement aléatoire de chacun de ses atomes. Les paramètres sont alors ajustés coopérativement de manière à ce que la molécule déformée tende un peu plus vers la conformation voulue à chaque cycle d'apprentissage.

Conclusion et perspectives

Nous avons actuellement mis en place la minimisation de l'énergie de Lennard-Jones à propos de laquelle nous obtenons des résultats plutôt encourageants en termes de temps de résolution. Les conformations finales trouvées ne permettent pas de conclure quant à la validité de l'algorithme étant donné que les fonctions ne sont pas connues à ce jour. Elle a en outre l'avantage par rapport à des méthode plus classiques telles que le recuit simulé ou les algorithmes génétiques, de n'emettre aucune hypothèse sur la fonction économique à minimiser. La prochaine étape consiste donc à mettre en place cet apprentissage et à le valider à l'aide de molécules connues. Elle sera réalisée et expérimentée dans les prochains mois. Toutefois, notre modélisation ne prend en compte ni le milieu (à savoir l'eau entourant nécessairement les molécules) ni les interactions électrostatiques dues à la polarité de certains atomes.
Cette approche est novatrice dans sa conception du repliement moléculaire et pourrait par la suite faciliter des découvertes sur l'implication de tel ou tel acide aminé dans le processus de repliement.

Références

BESSE C. (2003). Conformation de molécules par un système multi-agent adaptatif. In *Rapport de Stage Maîtrise, Institut Universitaire Professionnalisé Systèmes Intelligents, Université Paul Sabatier, Toulouse III*.

FAN E. (2002). Global optimization of lennard-jones atomic clusters. In *Thesis for the Degree Master of Science of McMaster University, Hamilton, Ontario*.

GEORGÉ J. P., GLEIZES M.-P. & GLIZE P. (2003). Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. In *Revue d'Intelligence Artificielle, RSTI série RIA, volume 17, n°4*, p. 591–626.

PDB T. (2003). The protein data bank. In P. BOURNE & H. WEISSIG, Eds., *Structural Bioinformatics*, p. 161–179.

A study of Amino Acids Binary Codes

Huaiguo Fu, Engelbert Mephu Nguifo

CRIL-CNRS FRE2499, Université d'Artois - IUT de Lens
Rue de l'université SP 16, 62307 Lens cedex. France.
E-mail:{fu,mephu}@cril.univ-artois.fr

Abstract :

If the biochemical properties of Amino acids can be perfectly described with certain binary codes, it can increase the potential of symbolic artificial intelligence methods to deal with protein folding problem. Thus we study four kinds of binary codes of amino acids (AA). Two codes of them are based respectively on biochemical properties, and the two others are generated with artificial intelligence (AI) methods, and are based on protein structures and alignment, and on Dayhoff matrix. In order to give a global significance of each binary code, we use a hierarchical clustering method to generate different clusters of each binary codes of amino acids. Each cluster is examined with biochemical properties to give an explanation on the similarity between amino acids that it contains. To validate our examination, a decision tree based machine learning system is used to characterize the AA clusters obtained with each binary codes. From this experimentation, it comes out that one of the AI based codes allows to obtain clusters that have significant biochemical properties. As a consequence, it appears that even if attributes of binary codes generated with AI methods, do not separately correspond to a biochemical property, they can be significant in the whole. Conversely binary codes based on biochemical properties can be insignificant when forming a whole.

This work allows to take into account biochemical properties of AA when binary codes are used to redescribe protein primary sequences.

Keywords: AI in Bioinformatics, Amino acids, Classification, Clustering

1 Introduction

A protein is typically built of a series of amino acids. Amino acids may come in a variety of shapes and properties: they may be small or bulky, hydrophobic or hydrophyllic, electrically charged or neutral, etc... hence allowing for very complex shapes and interactions to be produced. So the biochemical properties of amino acids are very important to analyse problems such as protein sequence alignment, protein secondary or tertiary structures prediction (Kawashima & Kanehisa, 2000; De la Maza, 1994).

Along with the development of bioinformatics, more and more methods and techniques of Artificial Intelligence (AI) are applied to solve problems of molecular biology such as protein secondary or tertiary structures prediction (Muggleton *et al.*, 1992). Such techniques are generally based on AA mutation matrices (Kawashima & Kanehisa, 2000). However there are a lot of symbolic AI techniques based on binary representations that could be applied in this domain. For example, genetic algorithms were used to predict protein structure (Unger & Moult, 1993; Pedersen & Moult, 1997). These works used single representation which does not catch many biochemical properties of AA. If the biochemical properties of AA can be perfectly described with certain binary codes, it will improve performance accuracy on the prediction of protein structure and function from its AA sequences (De la Maza, 1994). Hence in this paper we will focus on binary representation for AA. It is extended work of (Fu & Mephu Nguifo, 2004).

Expressing binary rules is more understandable for human-expert and could be helpful for providing efficient results explanation to the expert. Many symbolic AI systems deal with binary representations. They are unable to treat numerical values as that encodes in AA mutation matrices.

If several research works are being devoted to AA indices or mutation matrices (Kawashima & Kanehisa, 2000), our investigation of the literature gives rise only to four works on binary representation of AA : Dickerson & Geis (Dickerson & Geis, 1969), Marlière & Saurin (Sallantin *et al.*, 1984), De la Maza (De la Maza, 1994), and finally Gracy & Mephu (Mephu Nguifo, 1993).

De la Maza used primary and secondary structure of proteins as input, and combined genetic algorithm and neural networks algorithm, to generate the binary strings to represent AA. Gracy & Mephu applied a simulated annealing algorithm to Dayhoff's matrix to generate a binary representation of AA.

Dickerson & Geis and Marlière & Saurin proposed binary codes that are based on biochemical properties of AA. They consider different classification of biochemical properties of AA.

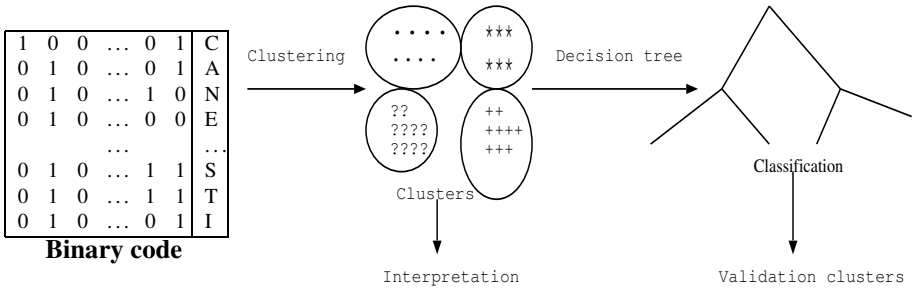


Figure 1: A view of the process of amino acids code comparison: clustering and decision tree.

In this study, we compare and analyse these 4 methods of AA binary representation (see a view of the process of amino acids code comparison in figure 1). In order to search for a global significance of each binary code, firstly, we use a clustering algo-

rithm to group AA. Then a machine learning system is used to explain the significances of the clusters using biochemical data. If the clusters perfectly correspond to certain biochemical properties of Amino acids, we consider this binary code as a good binary representation.

For our experimentation, we use an hierarchical clustering method (Ward's method (Lebart *et al.*, 1984)) to generate different clusters of AA. Each cluster is then examined by hand to give an explanation on the similarity between AA inside the cluster. To do this, we extend the representation of biochemical properties (hydrophobicity, charged, bulky, ...) of AA proposed by De la Maza (De la Maza, 1994). In order to validate and explain the clusters of AA, a public domain version of the decision tree based machine learning system C4.5 (Witten & Frank, 1999) is used to characterize the AA clusters obtained with the binary code.

The paper is organized as follows. The four AA binary representations are presented in next section. In the third section, we present hierarchical clustering to generate clusters of binary representations, and discuss the results obtained. And then a decision tree system C4.5 is used to validate the clusters of representations of AA in the fourth section.

2 Binary representations of amino acids

In this section, we describe the four AA binary representations. Binary representation of AA is a table of twenty rows and different columns. Each column is a property which can correspond to a biochemical property. Each row corresponds to an AA, and is a bit string where "0" means that this AA hasn't the property, and "1" means that this AA has the property.

2.1 Binary Codes based on biochemical properties

Two representations based on biochemical properties of AA are described by: Dickerson & Geis (Dickerson & Geis, 1969), and Marlière & Saurin (Sallantin *et al.*, 1984).

2.1.1 Dickerson & Geis's code

Dikerson & Geis's binary representation considers following properties of AA: aliphatic, aromatic, charged, polar, size of AA and hydrophobic (see figure 2). The table of the properties of AA could be easily transform to a binary representation.

Dickerson & Geis make an analysis of some protein sequences of the heavy and light chains of immunoglobulines to create this representation. A problem with this representation is that some AA have exactly the same physical and chemical properties in their classification, so the binary code of representation is the same, for example, A and G, I and L, Y and W, can't be distinguished. A way to solve this problem could be to add additional properties that allows to distinguish them.

Properties	List of AA
Hydrophobic	M I L V C A G T K H Y W F
Charged	D E K R H
Polar	Q N S C D E K R H Y W T
Positive	K R H
Small	P N D T V C S A G
Tiny	A G C S
Aliphatic	I L V
Aromatic	F Y W H

Figure 2: Classification of biochemical properties of amino acids proposed by Dikerson & Geis.

2.1.2 Marlière & Saurin's code

Marlière & Saurin propose to represent AA with 8 biochemical properties (Sallantin *et al.*, 1984) (see figure 3). On the basis of these 8 biochemical properties, each AA can be represented by a bit string like with the Dikerson & Geis's code. For example, we use 00010100 to represent the amino acid I (Isoleucine).

Properties	List of AA							
Side chain with less than 4 heavy atoms excluding H	A	C	G	P	S	T	V	
Side chain with more than 4 heavy atoms excluding H	E	F	H	K	Q	R	W	Y
Linear side chain	A	C	G	K	M	S		
Bulky side chain	F	H	I	P	T	W	Y	
Side chain with an oxygen	D	E	N	Q	S	T	Y	
Side chain without Z-H(Z=N, O, S) bond	A	F	G	I	L	P	V	
Charged side chain	D	E	H	K	R			
Side chain with less than 3 Carbon-Hydrogen bonds	C	D	G	N	S			

Figure 3: Biochemical properties of amino acids proposed by Marlière & Saurin.

This coding is a topologic description of AA. Pingand (Pingand, 1990) show that in such a coding appear sharply particular choices to certain types of studies, because certain criteria such as hydrophobicity in particular are debatable. This coding can be spread with the addition of other properties such as hydrophobicity, hydrophilic, etc. . . .

With such coding, it is also necessary to explicitly or implicitly add negation of properties in order to avoid inclusion between two AA codes. For example, the code of L (respectively R) is included into the code of I (respectively K).

2.2 Binary codes based on AI methods

De la Maza's (De la Maza, 1994) and Gracy & Mephu's (Mephu Nguifo, 1993) codes apply AI algorithms to generate binary representations. They propose a complete system to generate and test the binary representations of AA.

They use different techniques, but the whole structure is the same (see figure 4). In order to generate best binary representations, they use searching algorithms to find the best solution for representation of AA, from some data of AA or protein (e.g. Dayhoff's matrix, primary and secondary structure of protein).

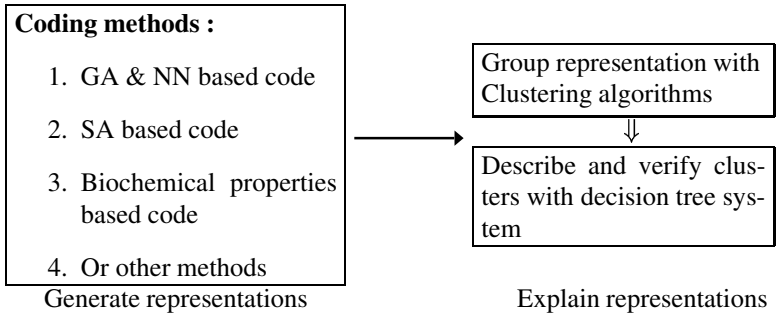


Figure 4: The system structure of the methods based on AI methods.

2.2.1 De la Maza's binary code of amino acids

De la Maza used the primary and secondary structure of proteins to create AA representations that facilitate secondary structure prediction (see figure 5). A genetic algorithm searches the space of AA representations. The quality of each representation is quantified by training a neural network to predict secondary structure using that representation. The genetic algorithm then uses the performance accuracy of the representation to guide its search and to create AA representations (see an example in figure 2.2.1) that improve the performance accuracy.

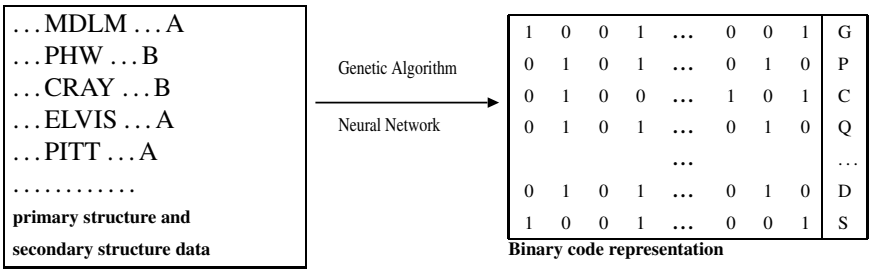


Figure 5: Generating the representations of amino acids with De la Maza's system.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	AA
1	0	0	1	0	1	1	1	1	0	1	1	1	0	0	0	Alanine
1	0	0	1	1	0	0	1	0	1	0	0	1	1	0	0	Cysteine
1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	Aspartic
0	0	1	1	1	1	1	0	1	1	0	1	1	1	1	0	Glutamic
0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	1	Phenylalanine
1	0	0	1	0	0	1	0	1	1	0	0	1	1	0	1	Glycine
1	1	1	0	1	0	0	0	1	0	0	0	0	0	1	1	Histidine
0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	Isoleucine
1	1	0	1	0	0	0	1	1	0	0	0	0	0	1	0	Lysine
1	0	1	1	1	1	0	1	0	1	1	1	1	0	0	1	Leucine
1	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	Methionine
1	0	0	1	1	1	1	0	1	1	0	0	1	0	0	1	Asparagine
0	1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	Proline
0	0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	Glutamine
0	1	0	1	0	0	0	1	0	0	1	1	0	0	1	1	Arginine
0	1	0	0	1	1	1	1	0	1	1	1	1	0	1	1	Serine
1	1	0	0	0	1	0	0	1	0	0	0	1	1	0	1	Threonine
1	1	0	1	1	1	0	0	1	1	1	0	0	1	1	1	Valine
0	0	0	0	1	1	0	0	1	1	1	1	0	1	1	0	Tryptophan
1	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	Tyrosine

Figure 6: 16 bits representations of de la Maza's code.

De la Maza (De la Maza, 1994) describes a system that synthesizes regularity exposing attributes from large protein databases. After processing primary and secondary structure data, this system discovers an AA representation that captures what are thought to be the three most important AA characteristics (size, charge, and hydrophobicity) for tertiary structure prediction.

2.2.2 Gracy & Mephu's binary code of amino acids

Gracy & Mephu's method uses the Dayhoff matrix and simulated annealing algorithms to generate the binary code of representation of 20 AA (see figure 2.2.2).

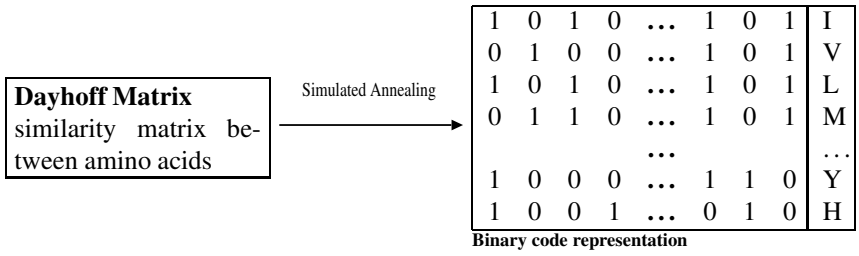


Figure 7: Generating the representations of amino acids in method of Gracy & Mephu.

Dayhoff (Dayhoff, 1972) proposed the matrix of similarity (probability of mutation) for AA. This matrix is a reference of major work of molecular biology, especially with the protein alignment problem or the protein structure prediction. Gracy & Mephu translate this similarity matrix into a distance matrix between AA, then use a simulated annealing algorithm to generate an AA binary representation that allows to approximate such distance matrix with the Euclidian distance measure.

Simulated annealing (Kirpatrick *et al.*, 1983) is a stochastic computational technique for finding near globally-minimum-cost solutions to large optimization problems. Some researches and applications have shown that simulated annealing is a technique which has a high probability of finding the optimal or a near-optimal solution in a reasonable time.

Using this method, we can get different representations of AA by changing the parameters of algorithms of this method. For example (see figure 8), we report the best 24-bits representation for 20 AA (see (Mephu Nguifo, 1993)).

G	1	0	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	1	0	1	0	0	1
P	1	0	0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0	1	0
C	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1	0	1
A	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1	0	1
N	1	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0
Q	1	0	1	0	0	1	1	0	1	0	1	0	1	0	0	1	0	1	1	0	1	0	1	0
E	1	0	0	1	1	0	1	0	1	0	1	0	0	1	0	1	1	0	1	0	1	0	1	0
D	1	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	0	1	0	1	0
S	1	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1
T	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1
I	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	1
V	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	0	1
L	1	0	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1
M	1	0	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1
F	1	0	1	0	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1
Y	0	1	1	0	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0
W	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	0	1	0	0	1	0	1	1	0
H	0	1	1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1	0
K	0	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1	1	0	0	1	1	0	1	0
R	1	0	1	0	0	1	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	0

Figure 8: 24 bits amino acids representation generated by Gracy & Mephu's method

.

3 Clustering analysis of binary representations of amino acids

In the previous section, four methods to represent AA with binary codes are presented. However, we face some questions: What's the significance of each binary representa-

tion? Which is a good AA representation?

To answer these questions, we use a clustering algorithm and decision tree system to validate the AA binary representations.

We use a clustering method to capture similarities among AA. This is similar to work done by de la Maza (De la Maza, 1994) to validate its binary code. This work is an extension to other binary representations.

3.1 Hierarchical Clustering

Clustering is often used for discovering structure in data. Cluster analysis (Dervin, 1996; K.Chan, 1991) provides a description or a reduction in the dimension of the data. In our work, clustering analysis is used for interpretation of binary representation of AA.

We use different hierarchical clustering methods available inside the SAS datamining package (SAS, 2001): Ward's method, Average Linkage method, and Centroid method. We focus our interest on the Ward's method (Lebart *et al.*, 1984) since different tries with these clustering methods generates the same clusters on each of the binary codes.

The Ward's method is an agglomerative hierarchical method which starts with each object describing a cluster, and then combines them into more inclusive clusters until only one cluster remains. The aim of the Ward's method is to unify groups such that the variation inside these groups is not increased too drastically.

For each binary code, we use the clustering algorithm to generate various number of clusters from 4 to 9. For example, with the 24 bits representation of *Gracy & Mephu's* method, when the number of clusters is 5, the result obtained is:

Cluster 1 : Asp, Glu, His, Lys, Asn, Pro, Gln, Arg.

Cluster 2 : Gly, Met, Val.

Cluster 3 : Cys, Ser, Thr.

Cluster 4 : Ala, Ile, Leu.

Cluster 5 : Phe, Trp, Tyr.

For our experimentation, we use the 8 bits code of Dikerson and Geis, the 16-bits representation of Marlière and Saurin (adding negation of each property), the best 16-bit representation of de la Maza, and the best 24-bit code reported by Gracy and Mephu. We obtain 24 sets of clusters, each set corresponding to a binary code and a fixed number of clusters.

Each cluster is then examined by hand to give an explanation on the similarity between AA inside the cluster.

3.2 Examination of clusters

In order to search for some significance of the binary codes of AA representations, we first examine by hand and analyze each cluster that we have obtained, with biochemical properties of AA.

The AA biochemical properties are at the basis of the interpretation of clusters of binary representations. We modify and extend the representation of chemical properties of AA proposed by De la Maza (De la Maza, 1994) (see figure 3.2). Modifications and

extensions come from discussion reported in recent publications on biochemistry (Kruh, 1995; Delaunay, 1997). We add some properties such as the mass, the number of atoms, and the hydrophobicity scale.

As an example, the result of 5 clusters with the 24 bits representation of Gracy & Mephu, is well-adapted to biochemical conditions and these clusters of AA have a certain logic of biochemical affinity:

Phenylalanine, Tryptophane and Tyrosine : aromatic AA with cyclic side chain and no charged.

Alanine, Isoleucine and Leucine : AA with side chains aliphatic.

Cysteine, Serine and Threonine : the Threonine differs of Serine by a grouping methyl and the Cysteine differs of serine by the presence of one atom of sulfur in the place of the atom of oxygen.

Asp, Glu, His, Lys, Asn, Pro, Gln, Arg: are more hydrophilic.

Gly, Met, Val are hydrophobic.

HYB	ARO	SOU	CHA	NEU	BAS	ACI	HYL	PI	-COO	$-NH_2$	$-R$	MM	N.A	E.H	AA
y	n	n	n	y	n	n	n	6.02	2.34	9.69	X	89	13	+1.8	Ala(A)
n	n	n	y	n	y	n	y	10.76	2.17	9.04	12.48	174	26	-4.5	Arg(R)
n	n	n	n	y	n	n	y	5.40	2.00	8.80	X	132	17	-3.5	Asn(N)
n	n	n	y	n	n	y	y	2.98	2.09	9.82	3.86	133	16	-3.5	Asp(D)
n	n	y	n	y	n	n	y	5.02	1.74	10.78	8.33	121	14	+2.5	Cys(C)
n	n	n	n	y	n	n	y	5.65	2.17	9.13	X	147	20	-3.5	Gln(Q)
n	n	n	y	n	n	y	y	3.22	2.19	9.67	4.25	146	19	-3.5	Glu(E)
y	n	n	n	y	n	n	n	5.95	2.34	9.60	X	75	10	-0.4	Gly(G)
n	n	n	y	n	y	n	y	7.59	1.82	9.17	6.00	155	20	-3.2	His(h)
y	n	n	n	y	n	n	n	6.05	2.40	9.70	X	131	22	+4.5	Ile(I)
y	n	n	n	y	n	n	n	5.98	2.36	9.60	X	131	22	+3.8	Leu(L)
n	n	n	y	n	y	n	y	9.74	2.18	8.95	10.53	146	24	-3.9	Lys(K)
y	n	y	n	y	n	n	n	5.75	2.30	9.20	X	149	20	+1.9	Met(M)
y	y	n	n	y	n	n	n	5.45	1.80	9.10	X	165	23	+2.8	Phe(F)
y	n	n	n	y	n	n	n	6.30	2.00	10.00	X	115	17	-1.6	Pro(P)
n	n	n	n	y	n	n	y	5.68	2.21	9.15	X	105	14	-0.8	Ser(S)
n	n	n	n	y	n	n	y	6.53	2.63	10.43	X	119	17	-0.7	Thr(T)
y	y	n	n	y	n	n	n	5.90	2.40	9.40	X	204	27	-0.9	Trp(W)
n	y	n	n	y	n	n	y	5.65	2.20	9.11	10.07	181	24	-1.3	Tyr(Y)
y	n	n	n	y	n	n	n	5.95	2.30	9.60	X	117	19	+4.2	Val(V)

Figure 9: The biochemical properties that are used for representation of AA. (y=yes, n=no, HYB= hydrophobic, ARO=aromatic, SOU=sulfur, CHA=charged, NEU=neutral, BAS= basic, ACI=acidic, HYL=hydrophlic, PI= pI value, molecular MM=mass, N.A=number of atom, E.H= hydrophobicity scale).

¿From the examination of all the sets of clusters obtained, it appears very often that there were always two or three clusters with debatable similarity, except for the previous one : set of 5 clusters of 24-bits representation of Gracy & Mephu.

With the coding of de la Maza, we didn't obtain the same clusters as that reported in (De la Maza, 1994). This may be due to the fact that de la Maza uses the Cobweb clustering algorithm which is different from the Ward's method.

With the coding based on biochemical properties, we were unable to find a set with good clusters.

From this first observation, it appears that coding based on AI method can have a

good global significance, whenever coding based on biochemical properties could not be significant in a whole. This global significance arises from the fact that similarity between AA is expressed inside the Dayhoff matrix in the case of Gracy and Mephu, or inside protein sequence alignment and protein structure in the case of de la Maza.

A second observation is made on properties of AI based coding. For the binary representation based on biochemical properties, each column corresponds to one biochemical property. For representations based on AI methods, we find that properties of binary codes do not separately correspond to biochemical properties.

Through clusters analysis and examination by hand, we obtain a global analysis of each binary representation. In the next section, we use decision tree system to validate our analysis.

4 Interpretation of clusters

In order to verify the clusters of binary representations of AA using biochemical data, we use a public domain of decision tree system C4.5 available in the WEKA package (Witten & Frank, 1999), to predict the cluster of an AA given its biochemical properties.

Using the clusters of representations and the database of biochemical properties shown in figure 3.2, we generate a decision tree to explain the clustering in terms of the biochemical properties. This decision tree is used to classify instances. Each node of the tree contains a test on an attribute. The branches that exit from a node correspond to the outcomes of the test. The leaves of the tree contain clusters.

¿From the results of decision tree, the representation based on AI methods can be shown that it corresponds to some biochemical properties of AA in varying degrees. It validates the accuracy of the clusters of AA representations.

However, even with the clusters and biochemical properties reported in de la Maza paper (De la Maza, 1994), we were unable to find the explanation tree obtained with the decision tree system as mentioned in his paper.

The results show that the clustering results of the 24-bits representation produced by Gracy & Mephu's method are correct and can be characterize with biochemical properties. This is in concordance with the results of our examinations by hand. This shows that the 24-bits representation produced by Gracy & Mephu's method is one of the best binary representations of AA. This corroborates previous results reported in (Mephu Nguifo, 1993; Landès *et al.*, 1996), where this representation allows to find good alignment when dealing with weakly homologous protein sequences.

For biochemical based representations, the decision tree can't give an understandable explanation of the clusters. This is in concordance with our analysis by hand. Thus biochemical properties based representations need to be preprocessed before being used, in order to express a whole significance.

¿From the results of clustering and decision tree process, we know that the methods of De la Maza, and of Gracy & Mephu based on AI methods can generate good binary representation of AA. These representations are significant in a whole, and allow to take into account biochemical properties when dealing with protein primary structure.

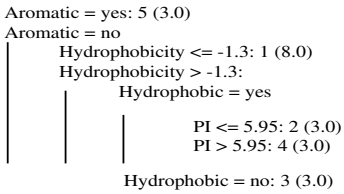


Figure 10: An example of decision tree(for 5 clusters of 24 bits representation of Gracy & Mephu).

5 Conclusion

This paper reviews two kinds of AA binary representations respectively based on biochemical properties, and on searching methods. A comparative study of these codes is described, and provides some significant results.

Properties of binary codes generated with AI methods, do not separately correspond to biochemical properties, but they are significant in a whole, conversely binary codes based on biochemical properties can be insignificant when forming a whole. And one of the codes proposed by Gracy and Mephu generates clusters that have significant biochemical properties. This fact seems to confirm a previous experimental result obtained with the alignment program, N+ONE (Landès *et al.*, 1996).

Good AA representations can facilitate the prediction of proteins secondary or tertiary structure, can allow to find good alignment of proteins primary sequences. This work could allow to improve results of AI methods when dealing with protein folding problem as it is well-established that data representation is one of the keys of success of AI methods.

References

DAYHOFF M. (1972). Atlas of protein sequence and structure, nat. *Biomedical Research Foundation,Washington DC*.

DE LA MAZA M. (1994). Generate, Test and Explain: Synthesizing Regularity Exposing Attributes in Large Protein DataBases. In *Proc. of the 27th Hawain Intl. Conf. on System Science (HICSS)*, p. 123–129, Hawa, USA.

DELAUNAY J. (1997). *Biochimie générale*.

DERVIN C. (1996). *Comment interpréter les résultats d'une classification automatique*.

DICKERSON R. & GEIS I. (1969). The structure and actions of proteins. *Harper & Row Publishers,New York, NY*, p. 16 – 17.

FU H. & MEPHU NGUIFO E. (2004). Clustering binary codes to express the biochemical properties of amino acids. In *The International Conference on Intelligent Information Processing (ICIIP)*.

KAWASHIMA S. & KANEHISA M. (2000). AAindex: Amino Acid index database. *Nucl. Acids. Res.*, **28**(1), 374–.

K.CHAN P. (1991). Machine learning in molecular biology sequence analysis.

KIRPATRICK S., GELATT C. & VECCHI M. (1983). Optimization by simulated annealing. *Science*, **220(4598)**, 671–680.

KRUH J. (1995). *Biologie cellulaire et moléculaire*.

LANDÈS C., BRAS F. & DEZÉLÉE, S.& TENINGES D. (1996). The gene 2 of the sigma rhadovirus genome encodes the p protein, the gene 3 encodes a protein related to the reverse transcriptases of retroelements. *Virology*, **215**, 123–142.

LEBART L., MORINEAU A. & WARWICK K. M. (1984). *Multivariate descriptive statistical analysis*. New York.

MEPHU NGUIFO E. (1993). *Concevoir une abstraction à partir de ressemblances*. Thèse de doctorat d'université, Université de Montpellier II. 276 pages.

MUGGLETON S., KING R. & STERNBERG M. (1992). Protein secondary structure using logic-based machine learning. *Protein Engineering*, **5**((7)), 647–657.

PEDERSEN J. & MOULT J. (1997). Ab initio protein folding simulations with genetic algorithms: simulations on the complete sequence of small proteins. *Proteins*, **S1**, 179–184.

PINGAND P. (1990). *Thesis of PhD*. PhD thesis, University of Montpellier.

SALLANTIN J., MARLIÈRE P. & SAURIN W. (1984). Description logique des contextes spatiaux dans les protéines: application à la conception de polypeptides artificiels. In *Actes des journées Point Curie sur l'intelligence artificielle et l'analyse des séquences biologiques*, p. 141–153, Paris, France: Institut Curie.

SAS (2001). *SAS Publishing, Getting started with enterprise miner*.

UNGER R. & MOULT J. (1993). Genetic algorithms for protein folding simulations. *Molecular Biology*, **231 (1)**, 75–81.

WITTEN I. H. & FRANK E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*.

Apprentissage d'automates par fusion de SFP (similar fragment pairs) et expérimentations sur les protéines MIP.

François COSTE et Goulven KERBELLEC
IRISA, Symbiose, Campus de Beaulieu, 35042 Rennes Cedex, France

Nous proposons une nouvelle approche heuristique permettant d'apprendre des automates non déterministes pour la caractérisation de familles de protéines. Cette approche est basée sur la fusion de paires de fragments significativement similaires (SFP). Elle ne nécessite aucun alignement multiple préalable. Notre heuristique utilise un ordonnancement d'une liste de SFP de tailles hétérogènes extraites du jeu d'apprentissage d'une famille de séquences protéiques. La souplesse de la méthode est apportée par différents types d'ordonnancement, notamment en fonction de la disponibilité, ou non, d'exemples de séquences n'appartenant pas à la famille à caractériser (contre-exemples). À partir d'une représentation du jeu d'apprentissage par un automate non déterministe, plusieurs étapes de généralisation nous permettent d'obtenir des motifs spécifiques à la famille. La généralisation s'effectuant par fusion des SFP, puis par unification des acides aminés non caractéristiques, la représentation par automate permet d'obtenir à la fois une localisation des zones conservées et une modélisation de l'enchaînement de ces zones. De plus, la prise en compte des propriétés physico-chimiques des acides aminés permet une étape supplémentaire de généralisation de certaines positions à des groupes de Taylor. La famille des MIP (Major Intrinsic Proteins) regroupe des séquences ayant la fonction de canal transmembranaire. Notre étude a pour objectif de caractériser la fonction des MIP, mais aussi celle plus précise, d'une sous-famille des MIP que l'on nomme water-specific (ex : AQP1, aquaporine) par opposition à la sous-famille non-water-specific (ex : GLPF, glycerol facilitator). Protomata-L, une implémentation de notre approche, génère des motifs qui montrent de bons résultats en rappel et précision, tant sur la caractérisation au niveau de la famille MIP que de la sous-famille Water-Specific. L'étude de la famille des MIP par notre approche fait apparaître la pertinence des automates appris. Une pertinence qui est bien établie pour le premier motif caractéristique des MIP en comparaison avec ceux de Prosite et de Pratt. La notion de coût, en terme d'erreurs nécessaires pour accepter une séquence dans un automate, est introduite par l'indice "error correcting cost". Il permet de faire ressortir également que la génération de motifs plus longs (de 40 à plus de 100 positions) et donc proches de la topologie des MIP permet une très bonne discrimination entre le jeu water-specific et le jeu non-water-specific.

Reconstruction supervisée de graphe génétique

Jean–Philippe Vert
Ecole des mines de Paris, Fontainebleau, FRANCE

A l'heure où les technologies dites "à heut débit" fournissent quantité de données sur les gènes et leurs produits, la reconstruction des interactions diverses entre ces composants se positionne comme un problème fondamental pour la compréhension du fonctionnement cellulaire au niveau systémique. Je présenterai une méthode, s'inspirant des récentes approches d'apprentissage statistique dans les espaces de Hilbert à noyau reproduisant, visant à reconstruire un graphe de gènes ou de protéines à partir de données sur les gènes, ainsi que d'une connaissance partielle du graphe à reconstruire. J'illustrerai cette approche avec une tentative de reconstruction des voies métaboliques et du réseau d'interactions protéiques de la levure Saccharomyces cerevisiae.

Élaboration de noyaux pour l'estimation de propriétés de petites molécules

Liva Ralaivola, Jonathan Chen, Jocelyne Bruand, Peter Phung, S. Joshua Swamidass et Pierre Baldi
Institute for Genomics and Bioinformatics, Université de Californie à Irvine

Les molécules comportant moins d'une centaine d'atomes et de liaisons chimiques jouent un rôle fondamental en chimie organique et biologie. Elles interviennent en effet dans plusieurs problématiques réelles comme celles, par exemple, de l'élaboration de principes actifs pour les médicaments. Le développement de grandes bases de données de tels composants chimiques soulèvent la question de la construction de méthodes automatiques capables d'en estimer les propriétés physiques, chimiques et biologiques.

Nous proposons plusieurs classes de fonctions noyau pour ces molécules de petite taille en exploitant leurs représentations 1D, 2D, et 3D. La représentation 1D repose sur une codage SMILE des molécules, la représentation 2D sur leurs graphes des atomes et liaisons et la représentation 3D sur les distances entre des paires prédéfinies d'atomes. Ces noyaux sont utilisés pour l'estimation de la mutagénicité, de la toxicité et de l'activité anti-cancéreuse de molécules issues de bases de molécules publiques et permettent d'obtenir des prédictions très fiables.