Clustering optimal de gènes fondé sur une mesure de similarité sémantique

Rachid HAFIANE* — **Malika SMAIL-TABBONE*** — **Marie-Dominique DEVIGNES**** — **Salvatore TABBONE***

**: Université de Lorraine, UMR LORIA 7503*
BP 239 LORIA - Campus scientifique
54506 Vandœuvre-lès-Nancy Cedex

***:CNRS, UMR LORIA 7503*
{rachid.hafiane,malika,devignes,tabbone}@loria.fr

RÉSUMÉ. *Dans de nombreux domaines d'application de l'analyse de données ou de la recherche d'information, il est utile de grouper de façon non supervisée des objets par similarité sans qu'il soit aisé de les représenter par des vecteurs de propriétés numériques. En biologie moléculaire, la similarité permet de capturer soit la structure complexe des objets (cas des molécules ou des séquences) soit la sémantique de leur description (cas des maladies ou des gènes). De nombreuses mesures de similarité ont été proposées mais une fois converties en dissimilarité en vue du clustering, ces mesures ne présentent pas forcément les bonnes propriétés d'une métrique. Le clustering d'objets pour lesquels on ne dispose que d'une matrice de dissimilarité requiert d'utiliser des méthodes adéquates. Nous proposons ici une évaluation comparative du clustering de gènes sur la base d'une mesure de similarité sémantique sur les termes de la Gene Ontology, IntelliGO. Nous nous appuyons sur quatre benchmarks que nous avons définis pour comparer les performances du clustering hiérarchique ascendant, du clustering C-means flou, et du clustering après plongement de la matrice de dissimilarité dans un espace Euclidien. Nous utilisons précisément une méthode de plongement qui tient compte du fait que la dissimilarité n'est pas une vraie métrique.*

ABSTRACT. *In various application domains of knowledge extraction or information retrieval, objects are not represented as feature vectors in a vector space but as a pairwise similarity matrix. In molecular biology, such a similarity measure either captures the object structure (e.g. molecules, proteins as sequences of amino acids) or the semantics of their description (genes or diseases described with ontology terms). The numerous existing similarity measures often violate metricity properties. This is the case of our IntelliGO semantic similarity defined as a generalized cosine between two vectors of Gene Ontology terms (Gene Ontology is a directed acyclic graph representing the semantic relationship between terms). Specific techniques exist*

for embedding pairwise data into Euclidian space for facilitating subsequent clustering of the objects. We report in this paper comparative gene clustering with and without embedding using the Intelligo measure and benchmarks. As for the clustering algorithm, we use an implementation of the C-means algorithm taking as input either a "distance" matrix or a set of vectors. We evaluate the clustering quality and discuss the results.

MOTS-CLÉS : *Mesure de similarité IntelliGO, dissimilarité non métrique, clustering hiérarchique ascendant, clustering, C-means flou, plongement*

KEYWORDS: *IntelliGO similarity measure, non metric dissimilarity, Hierarchical ascending clustering, Fuzzy C-means, Embedding*

1. Introduction et motivation

Dans de nombreux domaines d'application tels que la biologie moléculaire, où il est utile de réaliser une classification non supervisée ou clustering d'objets, il n'est pas aisé de représenter ces objets par des vecteurs de propriétés numériques. Les objets sont alors décrits par une matrice de similarité (Nugent *et al.*, 2010). En effet, cette similarité permet de capturer soit la structure complexe des objets, tels que des molécules ou des séquences, soit la sémantique de leur description comme cela peut être le cas pour des gènes ou des maladies. Lorsque l'utilisation d'une distance est nécessaire (cas de programmes de clustering), il est courant de convertir les valeurs de similarité (une fois normalisées) par une simple complémentation à 1. Or ce calcul ne permet pas toujours d'aboutir à une distance ayant les propriétés d'une métrique (en particulier l'inégalité triangulaire et l'indiscernabilité de deux objets dont la distance est nulle). Dans ce travail, nous appelons dissimilarité ce type de mesure et nous étudions leur comportement vis-à-vis de programmes de clustering.

La mesure de similarité IntelliGO que nous avons définie entre deux gènes (Benabderrahmane *et al.*, 2010) est un exemple qui ne fait pas exception puisque la dissimilarité correspondante viole l'inégalité triangulaire. Dans ce contexte, les gènes sont décrits à l'aide de termes appartenant à une ontologie dédiée, Gene Ontology ou GO (Consortium, 2010). GO est un graphe acyclique orienté (DAG) comprenant quelques 30 000 termes reliés par plusieurs types de relations sémantiques, principalement la spécialisation et la composition, et servant à décrire les processus biologiques dans lesquels le gène intervient, ses fonctions moléculaires et ses localisations cellulaires. Réaliser le clustering d'un ensemble de gènes sur la base de ses annotations GO permet d'identifier des groupes de gènes ayant des fonctions biologiques similaires. Cela est utile pour interpréter les résultats d'expériences à haut débit telles que la transcriptomique par exemple pour identifier parmi les gènes dérégulés chez des patients des groupes fonctionnels qui permettront de mieux comprendre les mécanismes liés à la maladie ou d'identifier des gènes marqueurs de différents stades de la maladie. Un clustering de gènes permet également la recherche des gènes responsables de maladies orphelines, la description en termes GO des symptômes de la maladie formant la requête.

Dans le cas d'une mesure de distance, il existe des méthodes de clustering adaptées telles que le clustering hiérarchique ascendant ou le clustering spectral. Ce dernier produit, grâce à un calcul de vecteurs propres, une représentation vectorielle des objets à partir d'une matrice de distances (Ng *et al.*, 2001). En revanche, dans le cas d'une mesure de dissimilarité, l'utilisation des méthodes précédentes devient hasardeuse. Roth et al. ont proposé une méthode constituant un raffinement du clustering spectral adapté à ce type de mesure grâce à un plongement spécifique de la matrice de dissimilarité dans un espace euclidien de représentation des objets initiaux (Roth *et al.*, 2003). Il devient alors légitime d'utiliser la plupart des algorithmes de clustering.

Nous proposons dans cet article de comparer les résultats du clustering de gènes en utilisant d'une part la matrice de dissimilarité obtenue sur la base de la mesure Intel-

liGO, d'autre part la représentation vectorielle obtenue après plongement des données de dissimilarité dans un espace euclidien. L'évaluation est faite sur la base de quatre benchmarks que nous avons définis afin d'évaluer divers algorithmes et mesures de similarité pour le clustering de gènes (Devignes *et al.*, 2012).

Nous présentons d'abord la mesure de similarité IntelliGO (section 2) puis les méthodes de clustering utilisées (section 3) et la méthode de plongement que nous avons sélectionnée (section 4). Enfin, nous décrivons les expériences de clustering, les résultats obtenus et une discussion de ces résultats (Section 5) avant de conclure.

2. Mesure de similarité Intelligo

Afin de réaliser le clustering fonctionnel de gènes, nous avons proposé une mesure de similarité nommée IntelliGO qui prend en compte les relations sémantiques entre les termes GO, le contenu d'information de ces termes (inversement proportionnel à leur fréquence dans l'ensemble des annotations de gènes), ainsi que la provenance de ces annotations (décrites dans la littérature, inférées automatiquement, etc.) (Benabderrahmane *et al.*, 2010).

La similarité entre deux gènes est définie comme le cosinus généralisé des vecteurs de termes en prenant en compte dans le produit scalaire la non-indépendance des termes.

Nous avons ainsi adapté la mesure du cosinus généralisé que Ganesan et al. avaient proposée pour les vocabulaires hiérarchiques (Ganesan *et al.*, 2003) à un vocabulaire structuré en DAG. La similarité entre deux termes t_i et t_j est définie en fonction de la profondeur maximale de l'ancêtre commun le plus spécifique (LCA, Least Common Ancestor) et du plus court chemin (SPL) entre les deux termes dans le DAG (Benabderrahmane *et al.*, 2010) :

$$Sim_{intelliGO}(t_i, t_j) = \frac{2 * profondeur(LCA)}{SPL(t_i, t_j) + 2 * profondeur(LCA)}$$

La similarité sémantique *IntelliGO* entre deux gènes g et h représentés par leurs vecteurs $\vec{g}$ et $\vec{h}$, respectivement est alors définie par :

$$SIM_{IntelliGO}(g, h) = \frac{\vec{g} * \vec{h}}{\sqrt{\vec{g} * \vec{g}}\sqrt{\vec{h} * \vec{h}}}, \quad [1]$$

Où :

– $\vec{g} = \sum_i \alpha_i * \vec{e_i}$: la représentation vectorielle du gène g dans l'espace vectoriel de la mesure *IntelliGO*.

– $\vec{h} = \sum_j \beta_j * \vec{e_j}$: la représentation vectorielle du gène h dans l'espace vectoriel de la mesure *IntelliGO*.

– α_i et β_j sont les coefficient des termes t_i et t_j pour les gène g et h respectivement, selon la mesure IntelliGO.

– $\vec{g}$*$\vec{h}$=$\sum_{i,j} \alpha_i * \beta_j$*$\vec{e_i} * \vec{e_j}$: représente le produit scalaire entre les deux vecteurs des gènes g et h.

– $\vec{e_i} * \vec{e_j} = Sim_{intelliGO}(t_i, t_j)$ telle que définie ci-dessus.

Notons que dans l'espace vectoriel associé à la mesure *IntelliGO*, les produits scalaires $\vec{g} * \vec{g}$ et $\vec{h} * \vec{h}$ ne sont pas réduits à la somme des carrés des coefficients des termes d'annotation des gènes g et h du fait de la non nullité des produits scalaires $\vec{e_i} * \vec{e_j}$.

Nous avons utilisé la mesure IntelliGO pour réaliser le clustering fonctionnel de gènes et avons montré qu'elle était robuste (Devignes *et al.*, 2012).

3. Algorithmes de clustering utilisés

$Fanny$ est le nom de l'implémentation dans le langage R de l'algorithme fuzzy C-means. $Fanny$ peut prendre en entrée soit un ensemble de vecteurs (et une distance au choix), soit une matrice de distances. La fonction objectif qui est minimisée dans ce dernier cas est la suivante :

$$\sum_{v=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} u_{iv}^{r} u_{jv}^{r} d(i,j)}{2 \sum_{j=1}^{n} u_{jv}^{r}}$$

où n est le nombre d'objets, k est le nombre de clusters, $d(i,j)$ est la distance entre les objets i et j, $u_{j,v}$ est la probabilité d'appartenance de l'objet j au cluster v. Le paramètre r permet de contrôler le chevauchement des clusters et est compris entre 1 (clusters non chevauchants ou partition) et 2 (clusters très chevauchants). Dans ce travail, étant donné que les ensembles de référence sont très peu chevauchants dans les benchmarks, la valeur de r choisie est celle qui se rapproche le plus de 1 et pour laquelle le clustering est possible (1.01 à 1.07 selon les cas). Quant à la valeur de k, elle variera entre 8 et 16 poue encadrer les nombres d'ensembles de référence des quatre benchmarks (13 pour les benchmarks 1 et 2 et 10 pour les benchmarks 3 et 4).

En ce qui concerne le clustering hiérarchique, nous avons utilisé l'algorithme de clustering hiérarchique ascendant (Jain *et al.*, 1988) implémenté par Kleiweg (http ://www.let.rug.nl/ kleiweg/clustering/) dans lequel le critère d'agglomération de deux clusters est fondé sur la distance moyenne entre toutes les paires d'objets provenant de chaque cluster.

4. Plongement de données de dissimilarité non conventionnelle dans un espace vectoriel

L'objectif d'un plongement ("embedding") est de construire, à partir d'une matrice carrée de dissimilarité ou de distance D de dimension n, une matrice de n vecteurs x_i dans un espace vectoriel de dimension p. Dans le cas d'une dissimilarité, une technique appropriée a été proposée dans (Roth *et al.*, 2003). Nous en présentons ici les principaux éléments.

On commence par centrer la matrice D. La matrice centrée lignes et colonnes d'une matrice P de taille $n \times n$ se calcule de la façon suivante. Soient e_n, un $n-$vecteur tel que $e_n = (1, ..., 1)^T$ et I_n, la matrice identité $(n \times n)$. Q est la projection dans le complément orthogonal de e_n, $Q = I_n - \frac{1}{n} e_n e_n^T$. La matrice centrée de P est notée P^c et définie par :

$$P^c = QPQ \qquad [2]$$

Après avoir centré D en D^c, on calcule la matrice centrée S^c selon l'équation (3).

$$S^c = -\frac{1}{2} D^c \qquad [3]$$

Pour que la décomposition en vecteurs propres corresponde à une matrice de distances euclidiennes, il faut que la matrice S^c utilisée soit positive semi-définie (dont les valeurs propres sont positives ou nulles). Comme ce n'est pas toujours le cas de S^c, on effectue un décalage constant avec la valeur propre minimale λ_{min} de S^c pour obtenir $\tilde{S^c}$ telle que :

$$\tilde{S^c} = S^c - \lambda_{min} I_n \qquad [4]$$

L'extraction des vecteurs propres peut alors se faire à partir de la matrice $\tilde{S^c}$ selon l'algorithme d'analyse en composantes principales (PCA). On a alors $\tilde{S^c} = V \Lambda V^T$ où $V = (v_1, v_2, ..., v_n)$ contient les vecteurs propres et Λ est la matrice diagonale des valeurs propres telle que : $\lambda_1 \geq ... \geq \lambda_p > \lambda_{p+1} = 0 = ... = \lambda_n$. On construit enfin la matrice $X_p = V_p (\Lambda_p)^{\frac{1}{2}}$ à partir des deux matrices V_p et Λ_p avec les p premières valeurs propres. C'est cette matrice X_p qui est utilisée ici pour le clustering. Notons que cette dernière étape est similaire à ce qui est réalisé dans le cas du clustering spectral directement à partir d'une matrice de distance qui n'a donc pas besoin d'être centrée ni décalée.

Voici sur un exemple de matrice de dissimilarité portant sur 6 objets le déroulement de la méthode de plongement :

```
Matrice carrée de dissimilarité D entre 6 objets :
```

```
 0.000 2.000 3.135 2.111 1.574 2.011
 2.000 0.000 4.604 0.091 1.330 3.546
 3.135 4.604 0.000 4.604 3.604 1.445
 2.111 0.091 4.604 0.000 1.230 3.520
 1.574 1.330 3.604 1.230 0.000 2.358
 2.011 3.546 1.445 3.520 2.358 0.000

Matrice D centrée :

 -1.546  0.331  0.496  0.444  0.150  0.124
  0.331 -1.792  1.841 -1.699 -0.216  1.536
  0.496  1.841 -3.732  1.844  1.087 -1.536
  0.444 -1.699  1.844 -1.787 -0.314  1.512
  0.150 -0.216  1.087 -0.314 -1.300  0.593
  0.124  1.536 -1.536  1.512  0.593 -2.229

Matrice S centrée (Sc) :

  0.773 -0.165 -0.248 -0.222 -0.075 -0.062
 -0.165  0.896 -0.921  0.849  0.108 -0.768
 -0.248 -0.921  1.866 -0.922 -0.544  0.768
 -0.222  0.849 -0.922  0.894  0.157 -0.756
 -0.075  0.108 -0.544  0.157  0.650 -0.297
 -0.062 -0.768  0.768 -0.756 -0.297  1.114

Valeur du décalage :  0.0 car lambda min de S centrée est égal à 0
Sc décalée identique à Sc

 Matrice des vecteurs propres :

  0.001 -0.734  0.527 -0.116  0.052 -0.408
 -0.439  0.285  0.192  0.187 -0.698 -0.408
  0.623  0.472  0.253 -0.397  0.003 -0.408
 -0.440  0.314  0.087  0.164  0.712 -0.408
 -0.182 -0.189 -0.691 -0.533 -0.056 -0.408
  0.438 -0.148 -0.369  0.695 -0.013 -0.408

 Matrice des valeurs propres :

 3.864 0.000 0.000 0.000 0.000 0.000
 0.000 1.060 0.000 0.000 0.000 0.000
 0.000 0.000 0.699 0.000 0.000 0.000
 0.000 0.000 0.000 0.530 0.000 0.000
 0.000 0.000 0.000 0.000 0.041 0.000
 0.000 0.000 0.000 0.000 0.000 0.000
```

```
Matrice X représentant les objets dans l'espace euclidien de dimension p=5 :

  0.002 -0.756  0.441 -0.084  0.002
 -0.864  0.294  0.161  0.136 -0.029
  1.225  0.486  0.212 -0.289  0.000
 -0.865  0.323  0.073  0.119  0.029
 -0.358 -0.195 -0.577 -0.388 -0.002
  0.860 -0.153 -0.308  0.506 -0.001
```

5. Évaluation du clustering de gènes avec et sans plongement

5.1. *Les benchmarks de groupes de gènes*

L'évaluation présentée dans cet article est faite sur la base de quatre benchmarks que nous avons définis avec soin afin d'évaluer divers algorithmes et mesures de similarité pour le clustering de gènes (Benabderrahmane *et al.*, 2011). Chaque benchmark comporte n groupes de gènes d'un organisme modèle (humain ou levure), chaque groupe partageant clairement un critère biologique (réseau biologique ou domaine fonctionnel). Les benchmarks sont décrits dans le tableau 1. Chaque groupe de gènes dans chaque benchmark est appelé ensemble de référence. Pour chaque benchmark, la matrice gènes$\times$ gènes de similarités IntelliGO est calculée puis convertie en matrice de dissimilarité par simple complémentation à 1 afin de permettre l'application des programmes de clustering et de plongement.

Benchmark	1	2	3	4
Organisme	Humain	Levure	Humain	Levure
Nombre d'ensembles de référence	13	13	10	10
Nombre total de gènes	268	168	94	118

Tableau 1. *Descriptif des quatre benchmarks.*

5.2. *Méthodes d'évaluation des résultats d'un clustering*

Lorsque des ensembles de référence sont disponibles, il existe plusieurs manières d'évaluer la qualité ou la validité d'un clustering par rapport aux ensembles de référence constituant la vérité terrain. Une première méthode est basée sur le calcul du F-Score (Van Rijsbergen, 1979). Cette méthode s'appuie sur l'appariement des clusters trouvés avec les clusters de référence et combine la précision et le rappel dans le calcul du F-Score. Le calcul du F-score global, ou $GFScore$ est présenté dans l'algorithme 1.

D'autres indices de comparaison de clusterings existent et nous en utilisons deux pour compléter le F-score d'après la compilation de (Wagner *et al.*, 2007). L'indice de Jaccard est défini comme la proportion de paires d'objets qui se trouvent dans le même cluster selon les deux clustering considérés. Soit R et C nos deux clusterings (référence et test), posons n_{11} : le nombre de paires d'objets présentes dans un même cluster à la fois dans R et C , n_{10} : le nombre de paires d'objets présentes dans un même cluster pour R et dans deux clusters différents pour C et n_{01} le nombre de paires d'objets présentes dans deux clusters différents pour R et dans le même cluster pour C . L'indice de Jaccard se calcule alors par :

$$J(R,C) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \quad [5]$$

Fondé sur la théorie de l'information, l'indice d'information mutuelle mesure la quantité d'information que la connaissance de la valeur de l'une des deux variables aléatoires exprimant l'appartenance d'un objet à un cluster dans l'un des deux clusterings nous apporte sur sa valeur dans l'autre clustering (Meilă, 2007). On définit d'abord pour chaque clustering C comportant l clusters son entropie par :

$$H(C) = -\sum_{i=1,l} P(i) * \log_2 P(i), \quad [6]$$

avec $P(i) = \frac{n_i}{n}$, représentant la probabilité pour un objet d'appartenir au cluster i qui contient n_i objets. On définit ensuite pour les deux clusterings à comparer R (comportant k clusters) et C (comportant l clusters), l'indice d'information mutuelle par :

$$I(R,C) = \sum_{i=1,k} \sum_{j=1,l} P(i,j) * \log_2 \frac{P(i,j)}{P(i) * P(j)}, \quad [7]$$

avec $P(i,j) = \frac{n_{ij}}{n}$, où n_{ij} est le nombre d'objets communs aux clusters i et j issus respectivement des classifications R et C. Pour des raisons évidentes de comparaison, on préfère utiliser une mesure normalisée de l'indice d'information mutuelle, par exemple selon la formule suivante :

$$MI(R,C) = \frac{I(R,C)}{\sqrt{H(R) * H(C)}}, \quad [8]$$

Ces indices de validité nous permettent aussi bien de comparer la qualité d'un clustering par rapport au clustering de référence que de comparer les résultats des trois méthodes testées.

Algorithm 1 Calcul du $GFScore$ entre un clustering et un clustering de référence (Benchmark)

Entrées : Deux clusterings R et C.
Sorties : La valeur du GFScore de C par rapport à R
  pour tout R_i classe de R **faire**
    pour tout C_j classe de C **faire**
      $precision(R_i, C_j) \leftarrow \frac{|R_i \cap C_j|}{|C_j|}$
      $rappel(R_i, C_j) \leftarrow \frac{|R_i \cap C_j|}{|R_i|}$
      $FScore(R_i, C_j) \leftarrow \frac{2 \times precision(R_i, C_j) \times rappel(R_i, C_j)}{precision(R_i, C_j) + rappel(R_i, C_j)}$
    fin
    $RFScore(R_i) \leftarrow \max_{C_j \in C} FScore(R_i, C_j)$
  fin
  $GFScore \leftarrow moyennePonderee_{R_i \in R} RFScore(R_i)$
  retourner $GFScore$

5.3. *Résultats et discussion*

Nous donnons, pour mémoire, dans le tableau 2 les valeurs du $GFScore$ que nous avions obtenues après le clustering hiérarchique sur les quatre benchmarks en faisant varier le nombre k de clusters entre 8 et 16 (Devignes *et al.*, 2012).

Nous avons lancé le programme $Fanny$ avec les quatre matrices de dissimilarité IntelliGO en faisant varier le nombre de classes k de 8 à 16. Les résultats des clusterings font ensuite l'objet du calcul du $GFScore$, de l'indice de Jaccard, et de l'information mutuelle.

D'autre part, le clustering des vecteurs résultant du plongement des matrices de dissimilarité correspondant à chaque benchmark est réalisée avec le même programme $Fanny$ (choix de la distance euclidienne). Le programme de plongement comporte un paramètre $rate$ qui permet de contrôler la taille des vecteurs dans la matrice X_p, c'est à dire le nombre de vecteurs propres considérés. La valeur de 100% utilisée ici revient à considérer tous les vecteurs propres. Les trois indices sont calculés sur les clusters obtenus.

Les valeurs des trois indices de validité obtenues avec le clustering $Fanny$ avant et après plongement pour les Benchmarks 1 et 2 sont reportés dans le tableau 3. Le tableau 4 contient les valeurs analogues pour les Benchmarks 3 et 4.

Les résultats des différents clusterings des quatre benchmarks peuvent être analysés selon plusieurs points de vue. Si l'on s'intéresse aux valeurs maximales des $GFScore$, on constate qu'elles sont beaucoup moins élevées avec le clustering hiérarchique (tableau 2) qu'avec le clustering $Fanny$ (tableaux 3 et 4), sauf pour le benchmark 3. Les mauvais résultats du clustering hiérarchique peuvent s'expliquer par la

K	Benchmark 1	Benchmark 2	Benchmark 3	Benchmark 4
8	50	37	61	48
9	50	43	62	59
10	50	44	**62**	**68**
11	51	49	68	68
12	52	49	76	68
13	**54**	**52**	76	68
14	55	55	76	68
15	55	56	82	68
16	55	56	81	68

Tableau 2. *Rappel des $GFScores$ (%) obtenus par clustering hiérarchique sur la base des matrices de dissimilarité IntelliGO*

	Benchmark 1						Benchmark 2					
	avant			après plongement			avant			après plongement		
K	GFS	$Jacc$	MI	GFS	$Jacc$	MI	GFS	$Jacc$	MI	GFS	$Jacc$	MI
8	63	39	70	68	45	73	69	42	78	72	45	81
9	67	43	71	70	47	75	74	49	79	76	51	81
10	64	37	70	70	44	75	79	55	83	82	59	86
11	67	39	71	78	57	80	80	60	85	85	66	89
12	73	47	75	82	59	83	81	57	85	83	59	88
13	**75**	**48**	**75**	**82**	**60**	**85**	**85**	**65**	**87**	**87**	**68**	**90**
14	68	40	73	69	42	76	87	68	89	87	65	89
15	65	35	70	76	50	81	83	56	86	86	64	90
16	70	41	73	77	47	79	16	9	0	79	52	87

Tableau 3. *Valeurs des trois indices de validité obtenues avec le clustering $Fanny$ avant et après plongement pour les Benchmarks 1 et 2. GFS = $GFScores$; $Jacc$ = Indice de Jaccard ; MI = Indice d'Information Mutuelle normalisé. Tous les scores sont exprimés en pourcentages.*

	Benchmark 3						Benchmark 4					
	avant			après plongement			avant			après plongement		
K	GFS	$Jacc$	MI	GFS	$Jacc$	MI	GFS	$Jacc$	MI	GFS	$Jacc$	MI
8	73	48	78	74	49	81	70	38	68	72	43	71
9	76	49	79	80	53	84	78	47	74	82	56	79
10	**76**	**47**	**81**	**74**	**44**	**82**	**75**	**45**	**73**	**80**	**50**	**75**
11	82	56	85	82	52	85	75	45	74	78	49	76
12	76	43	82	74	40	80	79	49	77	83	53	79
13	76	42	81	79	44	84	81	53	80	83	55	81
14	75	40	81	67	34	78	82	56	82	82	54	81
15	72	35	80	73	39	83	72	40	76	72	39	77
16	74	39	83	69	34	81	81	52	79	72	41	79

Tableau 4. *Valeurs des trois indices de validité obtenues avec le clustering $Fanny$ avant et après plongement pour les Benchmarks 3 et 4. GFS = $GFScores$; $Jacc$ = Indice de Jaccard ; MI = Indice d'Information Mutuelle normalisé. Tous les scores sont exprimés en pourcentages.*

rigidité de l'algorithme qui ne peut pas remettre en cause à l'étape n les agrégations effectuées à l'étape $n-1$.

Si nous comparons à présent les résultats obtenus avec le programme Fanny avec ou sans plongement, nous pouvons nous referer à la Figure 1 qui récapitule de façon graphique les valeurs des trois indices calculés pour chaque benchmark, pour des valeurs de K entourant le nombre d'ensembles de références : 13 pour les benchmarks 1 et 2 et 10 pour les benchmarks 3 et 4. Il apparait clairement que, à de rares exceptions près, les valeurs des trois indices sont concordantes et en général plus élevées lorsque le clustering est effectué après plongement. Ceci indique que le clustering réalisé après plongement de la matrice de dissimilarité dans un espace euclidien permet d'obtenir une classification plus proche de la vérité terrain.

Pour les benchmarks 1 et 2, les valeurs maximales des indices correspondent bien au nombre d'ensembles de référence. L'optimum est légèrement mieux marqué pour le clustering après plongement. Pour le benchmark 4, les valeurs obtenues semblent indiquer une meilleure classification en 9 clusters plutôt qu'en 10 avec comme précédemment un optimum mieux marqué après plongement. Le cas du benchmark 3 reste énigmatique car il ne présente pas d'optimum bien marqué ni de différence claire entre les valeurs obtenues avec ou sans plongement. Cette situation peut être liée à des particularités propres aux gènes contenus dans ce benchmark et montre la difficulté qu'il y a à travailler avec des données réelles. Cependant les résultats obtenus avec les trois autres benchmarks montrent assez clairement l'intérêt du plongement.

6. Conclusion

Nous apportons ici une réponse au problème du clustering d'objets représentés par une matrice de dissimilarité non conventionnelle dans laquelle la mesure de dissimilarité ne vérifie pas les propriétés d'une métrique. Trois méthodes ont été testées parmi lesquelles une alternative à l'utilisation directe de la matrice de dissimilarité consistant à introduire une étape de plongement destinée à transformer la représentation des données en une représentation vectorielle. Les résultats obtenus ici montrent clairement que l'introduction de cette étape de plongement en amont de l'algorithme de clustering conduit à une amélioration significative des performances du clustering sur la base de trois indices différents. Une étude plus approfondie est en cours pour étudier l'influence du paramètre $rate$ utilisé pour contrôler la taille des vecteurs en s'appuyant sur des approches classiques de sélection de variables afin d'optimiser encore le clustering.

Le passage à une représentation vectorielle permet d'utiliser en aval une grande variété d'algorithmes de clustering dont l'utilisation était jusqu'à présent hasardeuse voire impossible avec les mesures de dissimilarités atypiques, sémantiques ou spécifiques à certains objets biologiques, et ne vérifiant pas les propriétés d'une métrique.

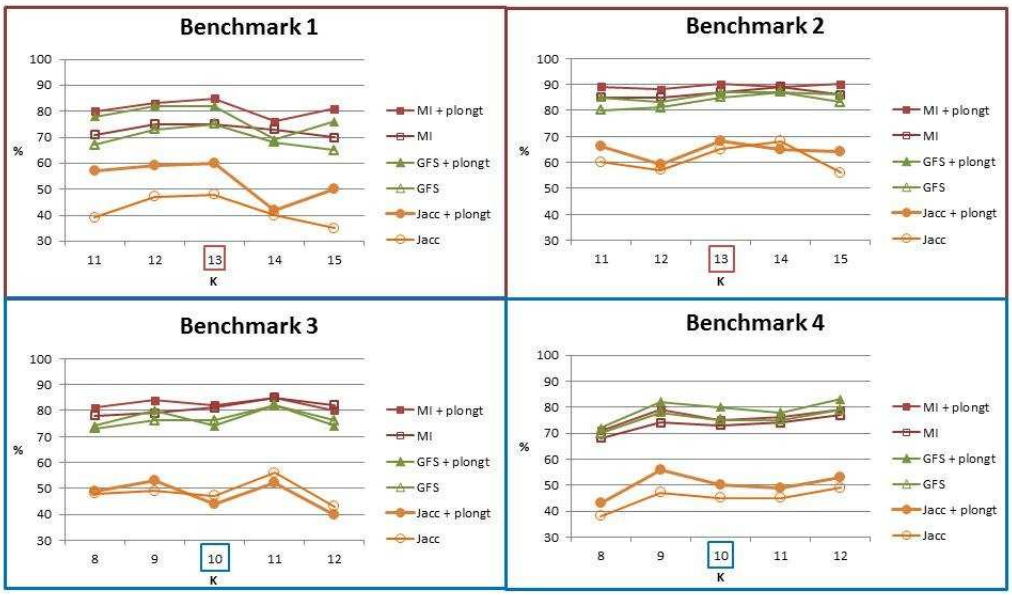

Figure 1. *Valeurs des trois indices calculés pour chaque benchmark, pour des valeurs de K entourant le nombre d'ensembles de références.*

7. Bibliographie

Benabderrahmane S., Devignes M.-D., Smaïl-Tabbone M., Napoli A., Poch O., « Ontology-based functional classification of genes : evaluation with reference sets and overlap analysis », *in* , Z. Zhao (ed.), *The Second Workshop on Integrative Data Analysis in Systems Biology - IDASB'11*, IEEE Computer Society, Atlanta, États-Unis, p. -, 2011.

Benabderrahmane S., Smaïl-Tabbone M., Poch O., Napoli A., Devignes M.-D., « IntelliGO : a new vector-based semantic similarity measure including annotation origin », *BMC Bioinformatics*, vol. 11/1, p. 588, 2010.

Consortium G. O., « The gene ontology in 2010 : extensions and refinements », *Nucleic Acids Reasearch*, vol. 38, p. D331-D335, 2010.

Devignes M.-D., Sidahmed B., Smail-Tabbone M., Amedeo N., Olivier P., « Functional classification of genes using semantic distance and fuzzy clustering approach : Evaluation with reference sets and overlap analysis », *international Journal of Computational Biology and*

Drug Design. Special Issue on : "Systems Biology Approaches in Biological and Biomedical Research", vol. 5/3, p. 245-260, 2012.

Ganesan P., Garcia-Molina H., Widom J., « Exploiting hierarchical domain structure to compute similarity », *ACM Trans. Inf. Syst.*, vol. 21/1, p. 64-93, January, 2003.

Jain A. K., Dubes R. C., *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, 1988.

Meilă M., « Comparing clusterings—an information based distance », *J. Multivar. Anal.*, vol. 98/5, p. 873-895, 2007.

Ng A. Y., Jordan M. I., Weiss Y., « On Spectral Clustering : Analysis and an algorithm », *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, MIT Press, p. 849-856, 2001.

Nugent R., Meila M., « An Overview of Clustering Applied to Molecular Biology Statistical Methods in Molecular Biology », *in* , H. Bang, , X. K. Zhou, , H. L. Epps, , M. Mazumdar (eds), *Statistical Methods in Molecular Biology*, vol. 620 of *Methods in Molecular Biology*, Humana Press, Totowa, NJ, chapter 12, p. 369-404, 2010.

Roth V., Laub J., Kawanabe M., Buhmann J. M., « Optimal Cluster Preserving Embedding of Nonmetric Proximity Data », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25/12, p. 1540-1551, 2003.

Van Rijsbergen C., *Information Retrieval*, Butterworth, 1979.

Wagner S., Wagner D., Comparing Clusterings-An Overview, Technical Report n 2006-04, Universität Karlsruhe (TH), 2007.

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com