MASTER D'INFORMATIQUE FONDAMENTALE, SECONDE ANNÉE
2010-2011

An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks

Gary D Bader and Christopher WV Hogue
BMC Bioinformatics, 2003

Présenté par :
Sylvain PRIGENT

Lieu de soutenance : Ecole Normale Supérieure de Lyon
Composition du jury : Dr. Eric Thierry, Dr. Christophe Crespelle
Relecteurs : Mélaine Aubry-Kientz, Jean-Marie Gomes

Soutenu le 24 novembre 2010

Table des matières

1 Présentation de MCODE

1.1 Présentation générale

De jour en jour, la quantité de données disponibles sur les interactions protéine-protéine et sur les complexes moléculaires s'accumulent dans les bases de données biologiques, notamment du fait des expérimentations de protéomique. La mise en place de méthodes de bio-informatique semble nécessaire à la prise en charge et à l'étude de ces données sans cesse plus nombreuses. C'est notamment le cas de la prédiction de l'existence de complexes protéiques à partir de données portant sur les interactions protéiques. Ce genre de prédiction revêt notamment une importance capitale dans l'objectif de réaliser une annotation fonctionnelle précise des protéines.

Bader et Hogue proposent ainsi une nouvelle méthode pour l'identification de complexes dans de grands réseaux d'interactions protéiques, appelée MCODE (Molecular COmplex DEtection). MCODE permettra également de faciliter la visualisation de grands réseaux protéiques en permettant d'extraire des régions "denses" autour d'une protéine d'intérêt.

Dans cette méthode, un réseau d'interactions protéiques est modélisé comme étant un graphe. Les sommets représentent les protéines et les arcs représentent les interactions entre celles-ci. Ces graphes peuvent être dirigés ou non dirigés selon le type d'interaction que l'on étudie et selon les informations dont on dispose. Ainsi, dans le cas de l'étude d'un signal cellulaire dont le sens est connu (en impliquant une cascade de phosphorylations par exemple), les flèches d'un graphe dirigé pourront représenter le sens de l'information. Dans le cas de la recherche de complexes protéiques, on va le plus souvent travailler avec des graphes non dirigés.

Une telle représentation des systèmes biologiques à l'aide de graphes permet l'utilisation de méthodes issues de la théorie des graphes afin d'analyser ces systèmes et de répondre à des questions biologiques en s'aidant de la puissance de calcul apportée par l'informatique.

MCODE va se baser uniquement sur le nombre de liens entre les sommets (ou protéines) pour former des clusters. Aucune pondération ne pourra être effectuée sur les arcs pour favoriser certaines interactions protéine-protéine. L'algorithme de MCODE peut être divisé en 3 étapes. Tout d'abord on va pondérer les sommets en fonction de leur densité locale de liens avec d'autres sommets. Ensuite vient la phase de prédiction des complexes. Cette prédiction sera éventuellement suivie d'une étape de post-traitement afin d'ajouter ou d'enlever des protéines de certains complexes selon certains critères de connectivité.

Pour effectuer la pondération des sommets, MCODE se base sur les travaux de Watts & Strogatz [1] qui introduisent un coefficient de regroupement C_i qui mesure la "cliqualité" du voisinage d'un sommet i. Le calcul des C_i se fait de la manière suivante :

$$C_i = \frac{2n}{k_i(k_i - 1)}$$

Avec : k_i : taille des sommets au voisinage du sommet i

n : nombre d'arcs au voisinage direct de i

Dans notre cas, la densité d'un graphe G = (V,E) est donnée par $D = \frac{|E|}{|E|_{max}}$ où $|E|$ représente le nombre d'arcs dans le graphe, et $|E|_{max}$ le nombre maximal d'arcs possiblement présents dans notre graphe, si celui-ci était une clique. La valeur de la densité du graphe est donc comprise entre 0 et 1. Posons $|V|$ le nombre de nœuds dans le graphe.

Si le graphe peut contenir des boucles, on a : $|E|_{max} = \frac{|V|(|V|+1)}{2}$

Si le graphe ne peut pas contenir de boucles, on a : $|E|_{max} = \frac{|V|(|V|-1)}{2}$

1.2 Pondération des sommets du graphe

La première étape de l'algorithme de MCODE consiste donc à pondérer tous les sommets du graphe. Pour cela on va pondérer les sommets en se basant sur la densité de leur réseau local. La densité du réseau local va être déterminée en utilisant les "k-core" du voisinage du sommet. Un k-core est un graphe possédant un degré minimal k. C'est-à-dire que tout sommet de ce graphe possèdera un degré supérieur ou égal à k. Par conséquent, le plus gros k-core d'un graphe correspond au sous-graphe le plus dense de ce graphe. Un autre terme est à définir : le coefficient de "core-clustering". Pour un sommet v, il s'agit de la densité du plus gros k-core au voisinage direct de v, v inclut. Le coefficient de core-clustering va avoir tendance à augmenter le poids des régions du graphe fortement interconnectées et à donner un poids plus faible aux sommets moins connectés, qui sont nombreux dans les réseaux d'interaction protéique sans échelle.

Le poids final donné à un sommet v, appelé $w(v)$, correspond au produit du coefficient de core-clustering avec le plus gros k-core au voisinage direct de ce sommet. Ce produit va avoir pour effet de maximiser le poids des nœuds densément connectés.

Pseudo code :

procedure MCODE-VERTEX-WEIGHTING

input : **graph** : G = (V,E)

for all v in G **do**

N = find neighbors of v to depth 1

K = Get highest k-core graph from N

k = Get highest k-core number from N

d = Get density of K

Set weight of v : $w(v) = k \times d$

end for

end procedure

1.3 Prédiction de complexes

La seconde étape de l'algorithme de MCODE correspond à la prédiction des complexes moléculaires. Elle nécessite d'avoir en entrée un graphe avec des nœuds pondérés.

Le sommet avec le plus gros poids va servir de graine à cette seconde étape. À partir de ce sommet, appelé "sommet-graine" on va aller récursivement vers l'extérieur, en incluant dans le complexe tous les sommets ayant un poids supérieur à un certain seuil qui sera proportionnel au poids du sommet-graine. Ce seuil est appelé paramètre VWP (Vertex Weight Percentage), noté d. Il est fixé pour une recherche donnée. Soit un sommet racine T et son voisin N. Le nœud N sera intégré dans le complexe enraciné par R si

$$w(N) > w(R) \times (1 - d)$$

A chaque fois qu'un sommet est inclut, tous ses voisins sont étudiés de la même manière pour voir s'il faut les inclure dans le complexe. Tous les sommets seront alors regardés une fois et une seule. Une fois que plus aucun sommet ne peut être ajouté à un complexe, on passe au sommet qui n'a pas encore été étudié possédant le plus haut poids. Les régions les plus denses du réseau vont être progressivement identifiées de cette manière. Le seuil utilisé pour choisir entre l'acceptation d'un nouveau nœud dans le graphe et son rejet va permettre de jouer sur la densité des sous-graphes (ou complexes protéiques) obtenus. Plus ce seuil sera proche du poids du sommet-graine, plus le sous-réseau autour de ce sommet sera dense.

Pseudo code :

procedure MCODE-FIND-COMPLEXES

input : **graph** : G = (V,E) ; **vertex weights** : W ;

vertex weight percentage : d

for all v in G **do**

if not already seen v **then call** : MCODE-FIND-COMPLEX(G, W, d, v)

end for

end procedure

procedure MCODE-FIND-COMPLEX

input : **graph** : G = (V,E) ; **vertex weights** : W ;

vertex weight percentage : d ; **seed vertex** : s

if s already seen **then return**

for all v neighbors of s **do**

if weight of v > (weight of s)(1 - d) **then** add v to complex C

call : MCODE-FIND-COMPLEX (G, W, d, v)

end for

end procedure

1.4 Optimisation des résultats

La dernière étape, bien qu'optionnelle, n'est pas moins très importante d'un point de vue pratique. Il s'agit du post-traitement des résultats. On commence par filtrer les "complexes protéiques" ne contenant pas, au moins, un 2-core.

L'option "fluff" va augmenter la taille des complexes selon un paramètre variant entre 0 et 1. Dans ce cas, étant donné un sommet *v*, ses voisins vont être ajoutés dans le complexe s'ils n'ont pas encore été vus et si la densité des voisins est plus grande que le paramètre de fluff choisi. Les sommets étudiés par cette étape ne sont pas marqués comme ayant été "vus", cela va permettre d'insérer une même protéine dans des complexes distincts, ce qui est en totale adéquation avec des données biologiques.

L'option "haircut" est également intéressante. En effet, après la recherche de complexes, certains nœuds qui étaient initialement reliés à plusieurs nœuds pourront se retrouver reliés au complexe par un seul arc.

L'utilisation de l'option "haircut" va supprimer ce type de sommets des complexes, de manière à n'avoir des complexes possédant au moins un 2-core.

Les complexes obtenus par cet algorithme sont évalués et classés. Le score d'un complexe correspond au produit de la densité du sous-graphe du complexe par le nombre de sommets du sous-graphe. Par conséquent, plus le score sera élevé, plus un complexe sera grand et dense.

Pseudo code :

procedure MCODE-FLUFF-COMPLEX
    input : **graph** : G = (V,E) ; **vertex weights** : W ;
        fluff density threshold : d ; **complex graph** : C = (U,F)
    for all u in C **do**
        if weight of u >d **then** add u to complex C
    end for
end procedure

procedure MCODE-POST-PROCESS
    input : **graph** : G = (V,E) ; **vertex weights** : W ; **haircut flag** : h ; **fluff flag** : f ;
        fluff density threshold : t ; **set of predicted complex graphs** : C
    for all c in C **do**
        if c not 2-core **then** filter
        if h is TRUE **then** 2-core complex
        if f is TRUE **then call** : MCODE-FLUFF-COMPLEX(G, W, t, c)
    end for
end procedure

Overall Process

procedure MCODE
    input : **graph** : G = (V,E) ; **vertex weight percentage** : d ;
        haircut flag : h ; **fluff flag** : f ; **fluff density threshold** : t ;
        set of predicted complex graphs : C
    call : W = MCODE-VERTEX-WEIGHTING (G)

call : C = MCODE-FIND-COMPLEXES (G, W, d)

call : MCODE-POST-PROCESS (G, W, h, f, t, C)

end procedure

Cet algorithme pourra également être utilisé avec des graphes orientés, en donnant en entrée un sommet-graine. Dans ce cas MCODE ne va tourner que pour prédire le seul complexe contenant le sommet-graine. Pour cela MCODE va préalablement étudier le réseau afin d'ignorer tous les sommets de poids supérieur à celui du sommet-graine. L'utilisation de graphes orientés semblent donc intéressant uniquement dans le cas où un premier traitement a été effectué avec un graphe non-orienté afin de découvrir tous les complexes possibles. Une fois cette étape effectuée, on pourra se concentrer sur un complexe particulier et y faire tourner l'algorithme avec des données de graphe orienté si celles-ci existent et sont intéressantes dans l'étude biologique.

2 Un exemple simple de l'utilisation de MCODE

Faisons tourner l'algorithme sur un exemple simple avec dix nœuds. La figure 1 donne le résultat obtenu et les deux "complexes protéiques" formés par MCODE avec un seuil de 0.2, à partir d'un tel graphe créé arbitrairement. Les nœuds entourés par des pointillés correspondent aux deux complexes identifiés par MCODE.

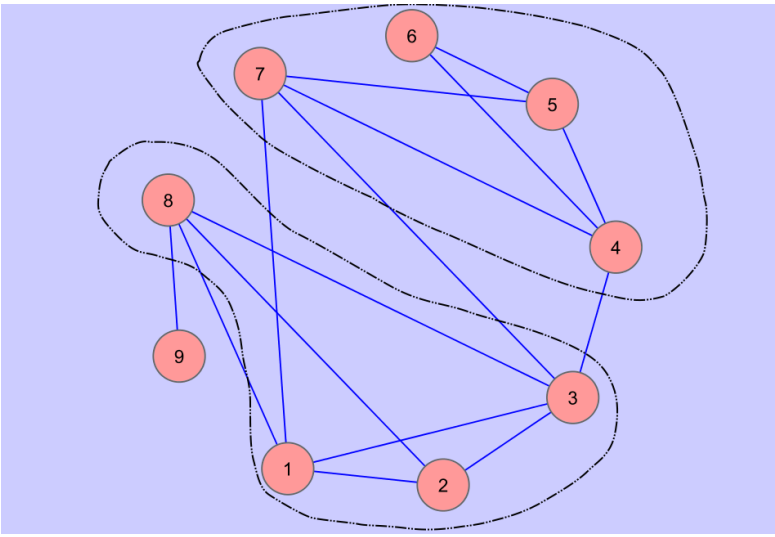

FIGURE 1 – Exemple de résultat obtenu après utilisation de MCODE sur un graphe simple

MCODE va commencer par assigner à chaque nœud un poids en utilisant la densité de son voisinage. Prenons par exemple le nœuds 5. Ses voisins sont les nœuds {4,5,6,7}. Le plus gros *k*-core formé par ce sous-graphe est un 2-core. Ici, la densité du nœud 5 vaut d = 5/6, on a en effet 5 arcs dans le sous-graphe formé par les nœuds {4,5,6,7} pour un maximum de 6 arcs si on avait eu une clique. Le nœud 5 possède donc un poids $w(5) = 2 \times 5/6 = 1.67$

MCODE suit le même raisonnement pour tous les nœuds et on obtient : w(1) = 3, w(2) = 3, w(3) = 3, w(4) = 1.4, w(5) = 1.667, w(6) = 2, w(7) = 1.4, w(8) = 3, w(9) = 1.

Une fois tous les nœuds pondérés, on passe à la prédiction de complexe proprement dite. Pour cela on choisit le sommet possédant le plus grand poids. Ici, quatre sommets ont un poids valant 3. Le nœud 1 peut être prit comme sommet-graine. Les voisins du nœud 1 seront intégrés dans le complexe s'ils ont un poids supérieur à $w(1) \times (1-d)$. Par défaut d vaut 0.2, on va garder cette valeur par défaut dans cet exemple. On va donc ajouter dans le complexe les voisins de 1 possédant un poids supérieur à $3 \times 0.8 = 2.4$, c'est-à-dire les nœuds 2, 3 et 8. On ne va pas intégrer le nœud 7 qui a un poids de 1.4 dans ce complexe. On effectue le même calcul pour tous nœuds que l'on a intégré dans le complexe, c'est-à-dire les nœuds 2, 3 et 8. Aucun des voisins de ces nœuds possèdent un poids suffisamment grand pour agrandir le complexe, on a donc un premier complexe formé par les sommets 1, 2, 3, 8. On continue ainsi avec un autre sommet-racine, c'est à dire le sommet possédant le plus gros score et qui n'a pas encore été intégré dans un complexe, le sommet 6. En procédant de la même manière que précédemment, on obtient un second complexe formé par les sommets 4, 5, 6, 7. Au final on a formé deux "complexes protéique" et on a une protéine qui est n'est incluse dans aucun complexe.

3 Complexité de l'algorithme

Pour étudier la complexité totale de l'algorithme on va étudier la complexité de chacune des parties.

Tout d'abord on recherche la présence de *k*-core dans le graphe. Cela va se faire en retirant progressivement les sommets et en ne gardant que les sommets possédant un degré supérieur ou égal à k. Au final on aura un ensemble de sommets reliés entre eux par un degré *k* ou supérieur. La complexité de cette étape est donc $O(n^2)$ avec n : nombre de sommets du sous-graphe. Trouver le plus grand *k*-core revient donc à effectuer cette recherche au pire n fois. On va donc avoir une complexité au pire en $O(n^3)$. Cette complexité est cependant à pondérer. En effet, ce n'est pas tous les sommets qu'il faut étudier mais seulement ceux au voisinage d'un

sommet donné. Si on appelle h le nombre moyen de sommets au voisinage direct d'un sommet dans le graphe, cette étape va alors avoir une complexité en O(h^3). Tout ceci va intervenir 2 fois pour chaque arcs contenu dans le graphe initial. Si on pose m, le nombre d'arc dans le graphe initial, on a donc une complexité en O($2mh^3$). Enfin cette étape de pondération aura lieu une fois pour chaque nœud du graphe. La phase de pondération dans sa totalité va donc avoir une complexité en O($2mnh^3$)

Une fois la pondération des sommets du graphe effectuée, MCODE va traverser le graphe pondéré avec un algorithme glouton afin d'isoler les régions densément connectées. On va donc avoir une prédiction des complexes en O(n).

La complexité de la phase de post-processing peut aller jusqu'en O(cs^2), avec c : nombre de complexes trouvés et s : nombre de sommets dans le plus gros complexe. Cette valeur correspond au temps nécessaire pour trouver des 2-core [O(s^2)] pour c complexes.

4 Discussion

MCODE est donc un moyen de rechercher des complexes moléculaires dans un réseau d'interaction protéique. Pour cela MCODE va rechercher des parties denses dans ce grand réseau d'interaction. Cette recherche de sous-graphes "denses" est un terrain d'intérêt qui a déjà été grandement étudié d'un point de vue théorie des graphes.

Ainsi, Hartuv et Shamir[2] ont proposé en 1999 un algorithme de clustering pour effectuer des recherches de sous-graphes denses dans des graphes. Celui-ci semble plus performant, avec une complexité totale en O($n^2 log(n)$). Cependant, il est à noter que dans le cas de MCODE, la grande majorité de la complexité réside dans le preprocessing et la pondération des sommets du graphe. Une fois cette étape effectuée, la prédiction des complexes se fait en O(n). Or la pondération n'a besoin d'être faite qu'une seule fois pour un graphe donné.

MCODE va donc permettre d'effectuer des recherches de complexes en faisant varier beaucoup de paramètres, pour coller au mieux à la biologie. Un autre des atouts de MCODE est qu'il ne va pas forcer la formation de complexes. Si un nœud (autrement dit une protéine) ne contient pas assez de lien avec les autres sommets du graphe, MCODE va la laisser de coté et ne va l'inclure dans aucun complexe, contrairement à la plupart des algorithmes de clustering qui vont avoir tendance à inclure tous les points d'un graphe dans des clusters. Cela n'est pas imaginable quand on considère des données biologiques telles que des interactions protéiques

et entrainerait la formation d'un grand nombre de faux positifs. Sachant que les expérimentations amenant à la connaissance d'interactions protéine-protéine portent intrinsèquement une forte propension à créer des faux positifs, il ne faut pas que le traitement de ces données crée encore plus de faux positifs.

5 Évaluation de MCODE

L'évaluation de MCODE va être faite sur des données biologiques vraies. MCODE va être ainsi lancé sur l'ensemble des données d'interaction moléculaires disponibles chez la levure *Saccharomyces cerevisiae*. Les complexes semblant porter un intérêt seront ensuite étudiés plus précisément avec le mode dirigé de MCODE.

Gavin et al.[3], en 2002, ont effectué des expériences de spectrométrie de masse à grande échelle pour mettre en évidence des complexes chez la levure. Á partir de 588 expérimentations, ils ont mis au jour 1363 protéines possiblement impliquées dans 3225 interactions. Ils ont également annoté manuellement 221 complexes protéiques. Ces complexes vont être utilisés pour évaluer MCODE, et pour fixer les paramètres à utiliser pour étudier les autres complexes, tels que le VWP ou le paramètre fluff. Cette analyse a apporté une sensibilité optimale de 0.31, et une spécificité optimale de 0.79. Si le chiffre de la spécificité est bon, on peut se demander pourquoi la sensibilité de MCODE est si faible. Cela tient en partie à la manière dont est considérée une "interaction" en entrée du graphe initial.

En effet, il est difficile d'étudier de manière précise l'interaction entre 2 protéines. Généralement pour savoir quelles protéines peuvent se fixer sur une protéine X donnée, on fixe X sur un support et on fait passer un ensemble de protéines sur X avant de passer à une étape de lavage pour supprimer les interactions non spécifiques. On obtient alors la protéine X fixée à son support, et plusieurs autres protéines fixées à X. On identifie alors ces protéines. imaginons que cette étape d'identification rapporte 3 protéines A, B, C. Il existe alors deux façons de voir les choses. Tout d'abord, il est possible de considérer que, si ces 4 protéines sont liées, elles le sont toutes deux à deux. Dans ce cas nous aurions les arcs suivants dans le graphe : {X-X, X-A, X-B, X-C, A-A, A-B, A-C, B-B, B-C, C-C}. Une autre manière de représenter ces interactions correspondrait à avoir les arcs suivants uniquement : {X-A, X-B, X-C, X-D}. Ce second exemple est appelé modèle Spoke. Bien que la meilleure des représentations se situe très probablement entre les deux représentations, le modèle Spoke a été utilisé pour représenter les données lors de l'évaluation de MCODE. Ce choix est bien sûr critiquable, mais un choix devait être fait. Et l'utilisation du premier modèle de représentation est totalement irréaliste dès que l'on a des complexes de grande taille, à cause notamment de l'encombrement stérique. De plus cette seconde

représentation risque d'apporter de très nombreux faux positifs.

La faible sensibilité est donc en grande partie due à cette représentation minimale des liens entre les protéines. Des études ultérieures et des recherches effectuées par cet outil ont montré qu'une bonne fixation des paramètres de fluff et d'haircut ainsi qu'une bonne définition des liens entre les protéines amènent à un accroissement significatif de la sensibilité.

6 Utilisation de MCODE

MCODE a été utilisé dans de nombreuses publications décrivant des réseaux d'interactions à grande échelle dans les journaux scientifiques les plus prestigieux. Par exemple, il a été utilisé pour étudier le réseau d'interaction protéique du parasite responsable de la malaria, *Plasmodium falsiparum* [4]. Dans cet exemple, MCODE a notamment servit à identifier un groupe de protéines impliquées dans la modification de la chromatine, la régulation de la transcription, la stabilité des ARN messagers et l'ubiquitination.

Web of Science recense 303 citations de l'article de Bader et Hogue en moins de 8 ans. Ces citations ont été effectuées dans des journaux portant sur des domaines aussi variés que la biologie cellulaire, l'informatique ou les mathématiques. Ceci est un signe indéniable de l'attrait de la communauté scientifique pour cet outil et pour la mise au point d'outils informatiques pour la biologie en général.

D'un point de vue pratique, MCODE est disponible en tant que plugin du logiciel Cytoscape[5].

Références

1. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393**(6684), 440–442, Jun (1998).
2. Hartuv, E. & Shamir, R. A clustering algorithm based on graph connectivity. *Information Processing Letters* **76**, 175–181 (1999).
3. Gavin, A.-C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A.-M., Cruciat, C.-M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M.-A., Copley, R. R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G. & Superti-Furga, G. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**(6868), 141–147, Jan (2002).
4. LaCount, D. J., Vignali, M., Chettier, R., Phansalkar, A., Bell, R., Hesselberth, J. R., Schoenfeld, L. W., Ota, I., Sahasrabudhe, S., Kurschner, C., Fields, S. & Hughes, R. E. A protein interaction network of the malaria parasite plasmodium falciparum. *Nature* **438**(7064), 103–107, Nov (2005).
5. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B. & Ideker, T. Cytoscape : a software environment for integrated models of biomolecular interaction networks. *Genome Res* **13**(11), 2498–2504, Nov (2003).